UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

Grado en ingeniería eléctrica

ESTUDIO DE LAS ETAPAS DE DISEÑO Y DESARROLLO DE UNA ARQUITECTURA CIBERFÍSICA PARA LA MONITORIZACIÓN DE UNA CELDA DE PRODUCCIÓN

MEMORIA

Trabajo de fin de grado

Proyectista: Albert Coll Guerra

Director/es: Miguel Delgado Prieto

Ángel Fernández Sobrino

Convocatoria: Enero 2020

I declare that,

the work in this Master Thesis / Degree Thesis (*choose one*) is completely my own work,

no part of this Master Thesis / Degree Thesis (*choose one*) is taken from other people's work without giving them credit,

all references have been clearly cited,

I'm authorised to make use of the company's / research group (*choose one*) related information I'm providing in this document (*select when it applies*).

I understand that an infringement of this declaration leaves me subject to the foreseen disciplinary actions by *The Universitat Politècnica de Catalunya - BarcelonaTECH*.

Albert Coll Guerra		13/01/2020
Student Name	Signature	Date

Title of the Thesis: Study of the stages of design and development of a cyber-physical architecture for the monitoring of a production cell. "Estudio de las etapas de diseño y desarrollo de una arquitectura ciberfísica para la monitorización de una celda de producción".

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

GLOSARIO

API: (*Application Programming Interface*) es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción. Son usadas generalmente en las bibliotecas de programación.

Bus: En general bus se refiere a un grupo de conductores paralelo que interconectan dos o más dispositivos en red.

Bus serie: Bus para la comunicación serie entre dos o más dispositivos.

DNS: (Domain Name System), es el método que se utiliza en la actualidad como forma sencilla de recordar los nombres de dominio en lugar de la IP a la que apuntan haciendo que los usuarios accedan más fácilmente a las webs.

HMI: (Human-Machine Interface), panel de control diseñado para conseguir una comunicación interactiva entre operador y proceso/máquina, con la función de transmitir órdenes, visualizar gráficamente los resultados y obtener una situación del proceso/máquina en tiempo real.

HTTP: (Hypertext Transfer Protocol), es el protocolo de comunicación que permite las transferencias de información en la World Wide Web.

IoT: (Internet of Things), es un concepto que se refiere a una interconexión digital de objetos cotidianos con internet.

Nodo: dentro de la informática la palabra nodo puede referirse a conceptos diferentes según el ámbito en el que nos movamos:

- En redes de computadoras cada una de las máquinas es un nodo, y si la red es Internet, cada servidor constituye también un nodo

- En estructuras de datos dinámicas un nodo es un registro que contiene un dato de interés y al menos un puntero para referenciar (apuntar) a otro nodo

PLC: (Programmable Logic Controller), Dispositivo programable encargado de la gestion automática de un proceso industrial. Para ver los PLCs utilizados ver Anexo III

RIO: (Remote Input Output), periferia de señales de campo compuesto por módulos de entrada y salida. Para ver los periféricos utilizados ver Anexo II

SQL: (Structured Query Language), es un lenguaje de dominio específico utilizado en programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales

VFD: (*Variable Frequency Drive*) o AFD (*Adjustable Frequency Drive*), es un sistema para el control de la velocidad rotacional de un motor de corriente alterna (AC) por medio del control de la frecuencia de alimentación suministrada al motor.

CAPÍTULO 1. INTRODUCCIÓN

1.1 Organización de la memoria

La memoria ha sido organizada de forma que el lector pueda entender y seguir con facilidad los pasos realizados. Todo está redactado para comprender los puntos siguientes, ya que se ha hecho referencia anteriormente.

En éste capítulo 1, introducción, hablaremos de la motivación que ha hecho posible este trabajo y definiremos sus objetivos propuestos en su día hasta el último de la mano del director del proyecto. Para terminar se hará una descripción general del trabajo por etapas.

En el capítulo 2 se presenta el estado del arte, se ponen en conocimiento las tecnologías involucradas directa o indirectamente en el proyecto que son necesarias para su interpretación y realización. Por ejemplo describiremos que tipos de cables se utilizan, cuales existen en el mercado o básicamente como se comunican los dispositivos entre ellos, aquí entrara un concepto nuevo “el protocolo”.

En el capítulo 3 y 4 se describe la arquitectura del proyecto realizado, diferenciando cada parte en cuestión, des de su hardware de estudio hasta el software creado, explicando con más detalle la forma en que se almacena y visualizan los datos obtenidos para su posterior análisis y mejora del proceso productivo.

Seguidamente en el capítulo 5 se hará mención a los procesos que se han llevado a cabo para entender el funcionamiento de los sistemas y en caso de no obtener los resultados oportunos como se han solventado.

Finalmente en el capítulo 6 hablaré de las conclusiones, aclaración de los conceptos más relevantes e impresiones personales o dificultades presentadas. También se hará inca pie en posibles mejoras o añadidos que se pueden implementar.

1.2 Motivación

La industria 4.0 en los últimos años está cogiendo mucho peso en el campo industrial, se está convirtiendo en una de las principales ventajas a nivel competitivo que diferencian unas empresas de otras. Hoy en día gracias a los avances tecnológicos y la gran reducción de precio en los dispositivos eléctricos y sobretodo electrónicos cualquier empresa de cualquier sector empresarial puede acceder a dicha tecnología.

Por este motivo, para optimizar el proceso productivo, rebajar los costes al máximo y tener en definitiva el mayor beneficio posible es necesario ayudarse de este saber hacer, esta nueva metodología de trabajo.

Como consecuencia dependiendo del sector en el que se trabaje o se trabajará, si se quiere iniciar un negocio empresarial, antes o después, las personas al frente de la empresa tendrán que sumarse a éste nuevo rumbo.

1.3 Objetivos

El objetivo de este proyecto consiste en el estudio y desarrollo de un sistema de recogida, almacenamiento, gestión y visualización a distancia de un proceso productivo de Schneider ubicado en las instalaciones de la UPC de Terrassa.

Dicho software de procesamiento y posterior gestión de datos estará compuesto por:

1. Obtención de datos del analizador de redes como tensiones y corrientes de línea para tener un histórico almacenado.
2. Obtención de datos que consideraremos importantes que estarán dentro del PLC almacenadas en sus marcas oportunas del programa.
3. Recolección y posterior almacenamiento en una base de datos.
4. Visualización y monitorización a distancia

1.4 Descripción del trabajo realizado

El proyecto se ha desarrollado en base a una celda de producción industrial de Schneider:

Esta celda de prueba, simulación o mejor dicho para caso de estudio ha planteado las siguientes etapas de realización:

1) Estudio del funcionamiento del programa junto a sus elementos físicos que intervienen para la automatización del proceso.

2) Diseño de la arquitectura ciberfísica del sistema para saber cuál, cuántos, como y donde conectar los dispositivos para su correcto funcionamiento.

3) Desarrollo del software del canal, servidor y visualizador gracias a la ayuda de los softwares Node-red, InfluxDB y Grafana.

CAPÍTULO 2. ESTADO DEL ARTE

2.1 Industria 4.0

"Industria 4.0 ha sido un término acuñado por el gobierno alemán con el soporte de industrias alemanas, para describir la digitalización de sistemas y procesos industriales, y su interconexión mediante el Internet de las cosas para conseguir una mayor flexibilidad e individualización de los procesos productivos. Es una visión de la fábrica inteligente. La transformación digital de la industria y las empresas con la integración de las nuevas tecnologías como Big Data, la Nube y la Ciberseguridad está produciendo el despliegue de la Cuarta Revolución Industrial". (Joyanes, 2017)

2.1.1 Evolución cronológica

La cuarta revolución industrial hace referencia a las cuatro fases de la revolución industrial:

- **Primera revolución industrial.** Máquinas de vapor y ferrocarril en el siglo XIX.
- **Segunda revolución industrial.** Motores eléctricos y producción en masa de principio del siglo XX. Aparece el motor de combustión, se desarrolla el aeroplano y el automóvil y, como grandes inventos aparece el teléfono y la radio.
- **Tercera revolución industrial.** Automatización y la informática en los años setenta del siglo XX.
- **Cuarta revolución industrial.** Los actuales sistemas ciberfísicos que recopilan y procesan información, toman decisiones inteligentes y ejecutan tareas en entornos cambiantes.

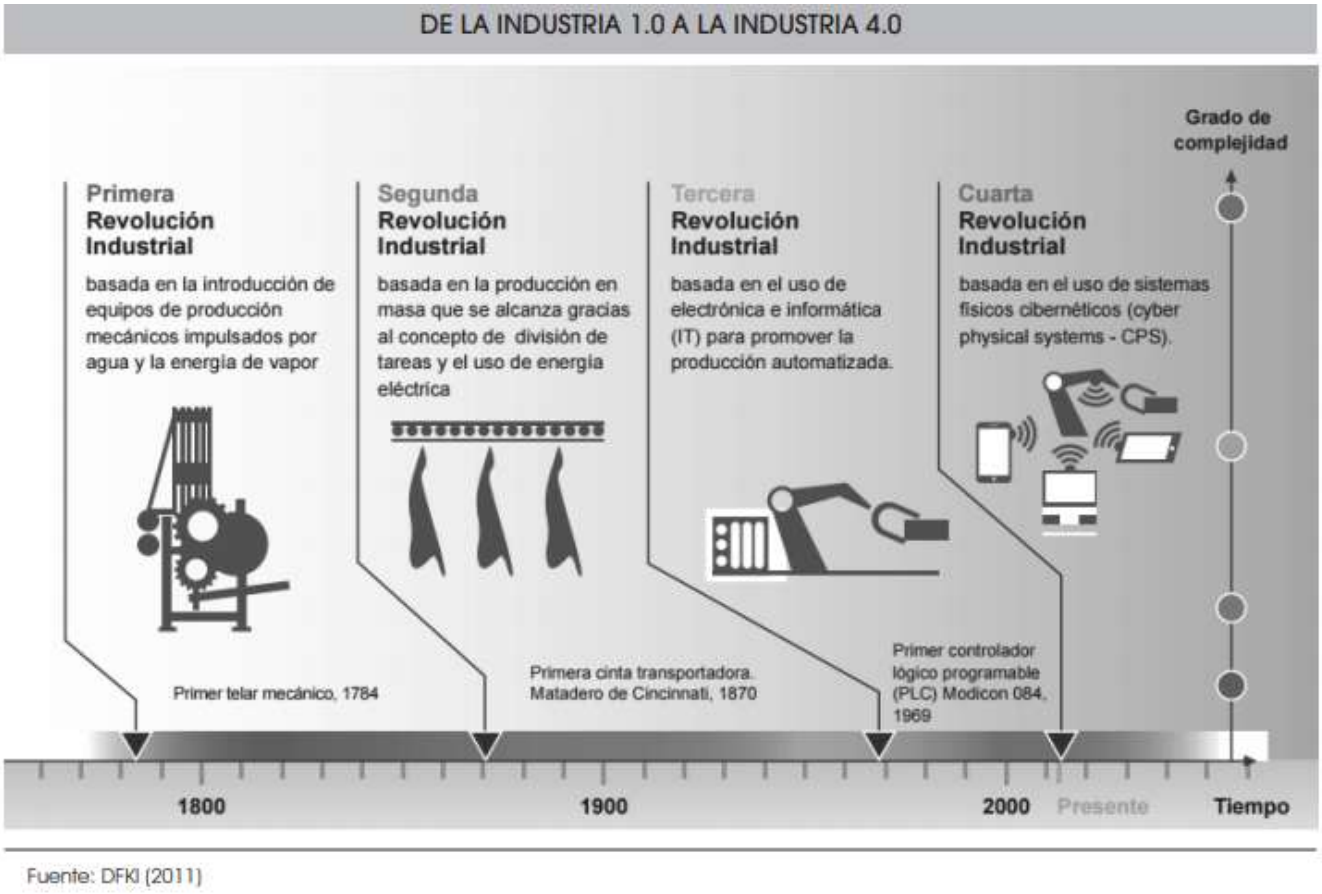


Figura 1 Las cuatro etapas de la revolución industrial

2.1.2 Actualmente

El nacimiento de la cuarta revolución industrial se confirmó en la Feria Tecnológica CeBIT 2015, uno de los eventos de referencia mundial y que también abordó esta cuarta revolución industrial. En la página principal de la feria se presentaron las tres fases clave a estudiar y desarrollar: siglo XIX (vapor y electricidad), siglo XX (computadoras/informáticas) y siglo XXI (digitalización). El tema central d!conomy y el país invitado, China, conformaron los objetivos fundamentales de la feria. Las cuatro tecnologías que se consideraron en CeBIT 2015 fueron: Datability, Big Data, Cloud, Movilidad y Social Business. Datability o nuevos medios para manipular y almacenar Big Data. Las tendencias se centraron en empresas star-ups, Big Data&Cloud, Movilidad, Seguridad IT, Social Business e Internet de las cosas. (Joyanes, 2017).

La era de la robotización no es cosa del futuro, es cosa del ahora

Muchas industrias del mundo desarrollado ya son parte de esta revolución, el uso de robots en sus operaciones de fabricación es una realidad que se está viviendo en este momento.

Ese proceso de avance no se detiene, es indetenible, las empresas de ayer no son las de hoy, es más, éstas se han debido adaptar a un fenómeno humano como es el aumento poblacional para cubrir la creciente demanda de productos de todo tipo; se ha pasado de una producción artesanal a la producción en serie, que hoy en día alcanza en la automatización una de sus últimas expresiones tecnológicas, con la creación de la inteligencia artificial que dota al robot de una capacidad nunca antes imaginada y que nos pone a las puertas de la llamada revolución de las máquinas para referirse a la robotización generalizada que interviene en cuanto proceso humano exista. Esa es la realidad que ya vivimos y que se intensificará con el transcurso de los próximos años. (Escuela de negocios, 2019).

INDUSTRIA 4.0 EN LA UE. ANÁLISIS DAFO

Fortalezas • Incremento de la productividad, de la eficiencia (recursos), de la competitividad y de los ingresos. • Aumento de los puestos de trabajo de alta calificación y muy remunerados. • Mejora de la satisfacción del cliente y nuevos mercados: incremento de la personalización de los productos y de su variedad. • Mayor flexibilidad y control de la producción.	**Debilidades** • Capacidad de adaptación tecnológica: pequeñas disrupciones pueden tener impactos grandes. • Dependencia de un abanico de factores de éxito: estándares, coherencia del entorno, oferta laboral con las habilidades apropiadas, inversión en I+D. • Costes de desarrollo y puesta en marcha. • Pérdida potencial de control sobre la empresa. • Puestos de trabajo semi-formados. • Necesidad de importar mano de obra formada e integrar los inmigrantes.
Oportunidades • Reforzamiento de la posición de Europa como líder en industria manufacturera y otros sectores. • Desarrollo de nuevos mercados punteros para productos y servicios. • Contrapunto a la demografía negativa de la UE. • Disminución de las barreras de entrada para algunas PYMES para participar en nuevos mercados y nuevas cadenas de suministro.	**Amenazas** • Ciberseguridad, propiedad intelectual, privacidad de los datos. • Trabajadores, PYMES, sectores y economías nacionales sin conciencia y/o medios para adaptarse a la Industria 4.0 y que quedarán atrás. • Volatilidad de las cadenas de valor globales y vulnerabilidad hacia ellas. • Adopción de la Industria 4.0 por parte de los competidores

Figura 2 Tabla DAFO

Fuente: Smit *et al.* (2016)

2.2 Fundamentos teóricos de la comunicación

Para entender de forma fácil cómo se comunican las máquinas las compararemos con las personas.

Como vemos en la imagen cuando nos comunicamos existe un emisor y un receptor y para que puedan entenderse utilizan un mismo código común, en este caso el idioma.

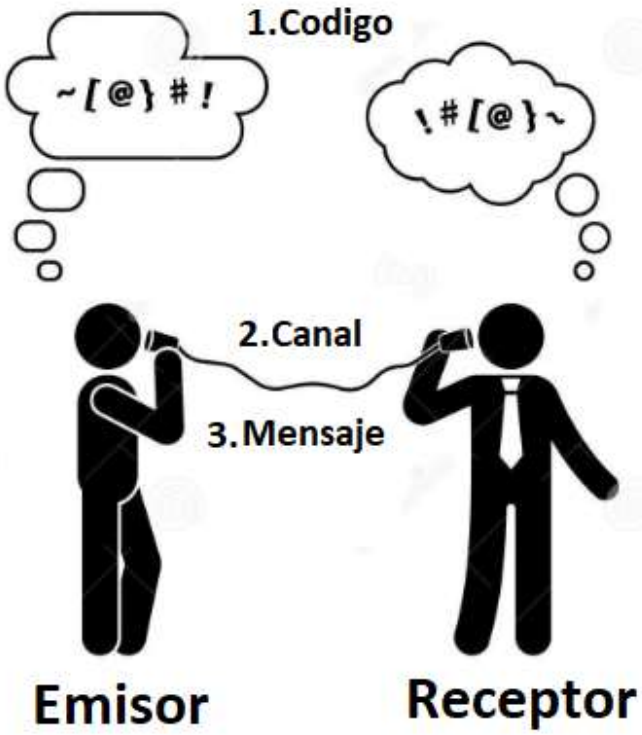


Figura 3 Comunicación humana

1. Conjunto de reglas y signos que comparten los datos

2. Medio físico por el que se transmite el mensaje

3. Información que se envía

Cuando hablamos de comunicación en redes industriales, el idioma es “el protocolo”.

Siempre que haya comunicación entre dos dispositivos quiere decir que tienen mismos protocolos.

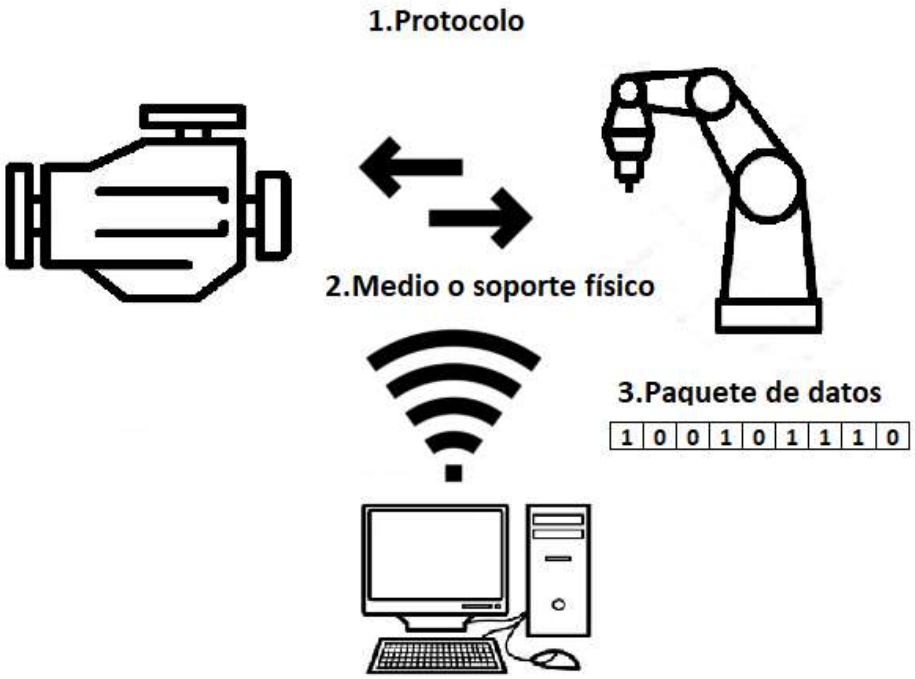


Figura 4 Comunicación entre dispositivos

2.3 Redes

2.3.1 Hardware de red

No existe una clasificación aceptada en la que encajen todas las redes, pero hay dos que sobresalen de manera importante: la tecnología de transmisión y la escala.

2.3.1.1 Tecnología de transmisión

Hablando en sentido general, existen dos tipos de tecnología de transmisión que se emplean mucho en la actualidad: los enlaces de **difusión (*broadcast*)** y los enlaces de **punto a punto**.

Los enlaces de punto a punto conectan pares individuales de máquinas. Para ir del origen al destino en una red formada por enlaces de punto a punto, los mensajes cortos (conocidos como paquetes en ciertos contextos) tal vez tengan primero que visitar una o más máquinas intermedias. A menudo es posible usar varias rutas de distintas longitudes, por lo que es importante encontrar las más adecuadas en las redes de punto a punto.

A la transmisión punto a punto en donde sólo hay un emisor y un receptor se le conoce como **unidifusión (*unicasting*).**

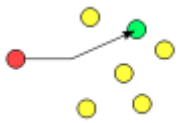


Figura 5 Grafico unicast

En una red de **difusión (broadcast)** todas las máquinas en la red comparten el canal de comunicación; los paquetes que envía una máquina son recibidos por todas las demás. Un campo de dirección dentro de cada paquete específica a quién se dirige. Cuando una máquina recibe un paquete, verifica el campo de dirección. Si el paquete está destinado a la máquina receptora, ésta procesa el paquete; si el paquete está destinado para otra máquina, sólo lo ignora.

Una red inalámbrica es un ejemplo común de un enlace de difusión, en donde la comunicación se comparte a través de una región de cobertura que depende del canal inalámbrico y de la máquina que va a transmitir.

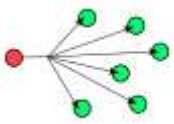


Figura 6 Grafico broadcast

Por lo general, los sistemas de difusión también brindan la posibilidad de enviar un paquete a *todos* los destinos mediante el uso de un código especial en el campo de dirección. Cuando se transmite un paquete con este código, todas las máquinas en la red lo reciben y procesan. A este modo de operación se le conoce como difusión (*broadcasting*). Algunos sistemas de difusión también soportan la transmisión a un subconjunto de máquinas, lo cual se conoce como multidifusión (*multicasting*).

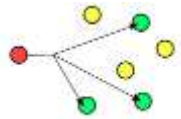


Figura 7 Grafico multicast

2.3.1.2 Escala

Hay un criterio alternativo para clasificar las redes: por su escala. La distancia es importante como medida de clasificación, ya que las distintas tecnologías se utilizan a diferentes escalas.

2.3.1.2.1 PAN

Las redes de área personal, generalmente llamadas PAN (*Personal Area Network*) permiten a los dispositivos comunicarse dentro del rango de una persona. Un ejemplo común es una red inalámbrica que conecta a una computadora con sus periféricos.

2.3.1.2.2 LAN

Las redes de área local, generalmente llamadas LAN (*Local Area Networks*), son redes de propiedad privada que operan dentro de un solo edificio, como una casa, oficina o fábrica. Las redes LAN se utilizan ampliamente para conectar computadoras personales y electrodomésticos con el fin de compartir recursos (por ejemplo, impresoras) e intercambiar información.

2.3.1.2.3 MAN

Una Red de Área Metropolitana, o MAN (*Metropolitan Area Network*), cubre toda una ciudad. El ejemplo más popular de una MAN es el de las redes de televisión por cable disponibles en muchas ciudades.

2.3.1.2.4 WAN

Una Red de Área Amplia, o WAN (Wide Area Network), abarca una extensa área geográfica, por lo general un país o continente.

2.3.1.3 Topología

"El término topología se refiere a la forma en que está diseñada la red, bien físicamente (rigiéndose de algunas características en su hardware) o bien lógicamente (basándose en las características internas de su software). La topología de red es la representación geométrica de la relación entre todos los enlaces y los dispositivos que los enlazan entre sí (habitualmente denominados nodos). Para el día de hoy, existen al menos cuatro posibles topologías de red básicas: malla, estrella, bus y anillo". (Vergara, 2007).

2.3.1.3.1 Malla

Power Cert (2017) afirma que cada dispositivo se conecta a cada uno de los dispositivos de la red. Por tanto teniendo tantas conexiones en caso de algún fallo se maneja muy bien. Debido a la cantidad de cables y conexiones es cara de implementar, por eso es rara en una LAN, son principalmente utilizadas en WLAN. El claro ejemplo seria el internet. Vergara (2007) añade que una malla ofrece varias ventajas sobre otras topologías de red. En primer lugar, el uso de los enlaces dedicados garantiza que cada conexión sólo debe transportar la carga de datos propia de los dispositivos conectados,

eliminando el problema que surge cuando los enlaces son compartidos por varios dispositivos. En segundo lugar, una topología en malla es robusta. Si un enlace falla, no inhabilita todo el sistema.

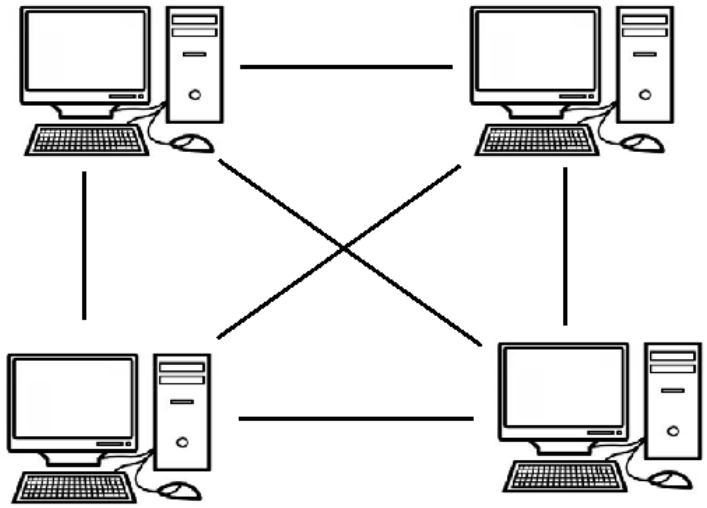


Figura 8 Topología malla

2.3.1.3.2 Estrella

"En la topología en estrella cada dispositivo solamente tiene un enlace punto a punto dedicado con el controlador central, habitualmente llamado concentrador. Los dispositivos no están directamente enlazados entre sí.

A diferencia de la topología en malla, la topología en estrella no permite el tráfico directo de dispositivos. El controlador actúa como un intercambiador: si un dispositivo quiere enviar datos a otro, envía los datos al controlador, que los retransmite al dispositivo final". (Vergara, 2017).

Según Power Cert (2007) es la más comuna. Como se ve en la imagen de abajo todos los dispositivos están cableados en un punto central. Todos los datos pasan a través de un punto central antes de continuar hacia su destino. Una de las desventajas que tiene esta

tipología es que en caso de que falle el dispositivo central todos los dispositivos se verían afectados y caerían.

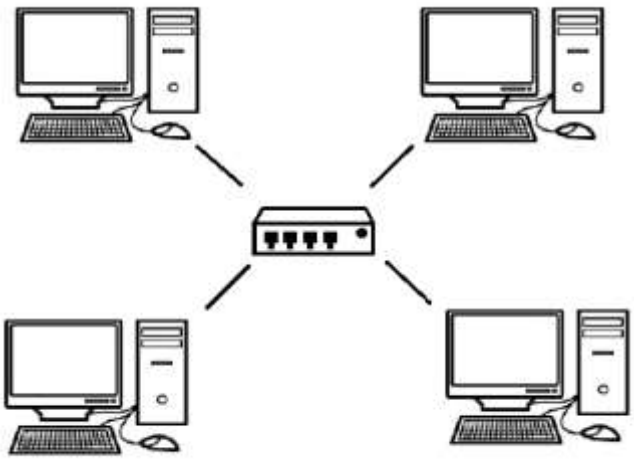


Figura 9 Topología estrella

2.3.1.3.3 Bus

Power Cert (2017) expone que es una tipología antigua y no se utiliza tanto hoy en día. Se caracteriza por tener un único canal de comunicaciones (denominado bus, troncal o backbone) al cual se conectan los diferentes dispositivos. De esta forma todos los dispositivos comparten el mismo canal.

Este tipo de dispositivos se caracterizan por tener una serie de ventajas. Por ejemplo, son muy económicos y son fáciles de implementar. Sin embargo, también agrupan algunas desventajas. En éste caso, son dispositivos que requieren que el cable termine con una terminal en ambos lados. Si se quita o agrega un dispositivo o si los terminadores de ajuste se pierden o faltan, entonces el cable estaría abierto y los datos rebotarían, esto se conoce como reflexión de señal. Si esto sucede, el flujo de datos se interrumpiría.

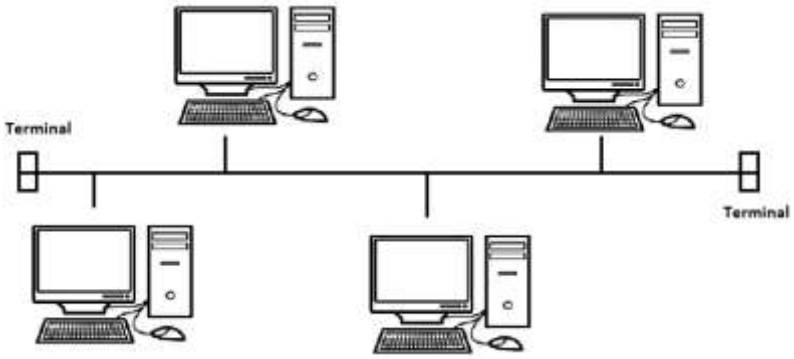


Figura 10 Topología bus

2.3.1.3.4 Anillos

Power Cert (2017) explica que cada dispositivo está conectado al otro y confinado en un bucle cerrado, por tanto cada dispositivo tiene dos vecinos para comunicarse. Cada paquete de datos es enviado alrededor del anillo hasta que llega a su destino final. Esta tipología es raramente utilizada hoy en día.

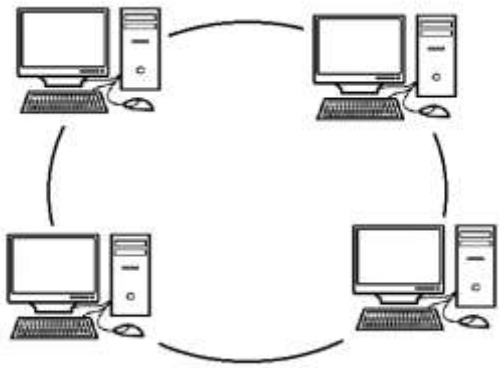


Figura 11 Topología anillo

"Un anillo es relativamente fácil de instalar y reconfigurar. Cada dispositivo está enlazado solamente a sus vecinos inmediatos (bien fisicos o lógicos). Para añadir o quitar dispositivos, solamente hay que mover dos conexiones. Las únicas restricciones están relacionadas con aspectos del medio fisico y el tráfico (máxima longitud del anillo y número de dispositivos). Además, los fallos se pueden aislar de forma sencilla. Generalmente, en un anillo hay una señal en circulación continuamente". (Vergara, 2007).

2.2.3.5 Topologías híbridas

Por último, Power Cert (2017) expone que las topologías vistas hasta ahora también se pueden combinar entre ellas para beneficiarse y ofrecer lo mejor de cada mundo. Las hibridaciones más comunes son:

- Estrella-Anillo: Dos o más topología en estrella se unen para formar una larga red en anillo.

- Estrella-Bus: Dos o más topologías en estrella se unen usando un solo bus de conexión.

2.3.1.4 Direccionalidad de los datos

2.3.1.4.1 Simplex

En el modo simplex, la comunicación es unidireccional, como en una calle de sentido único, de forma que una de las dos estaciones de enlace puede transmitir y la otra solo recibir (Forouzane, 2007)

Figura 12 Ilustración comunicación simplex

2.3.1.4.2 Semi-Duplex (Half-Duplex)

En este caso cada estación puede tanto enviar como recibir, pero no al mismo tiempo. Así, cuando un dispositivo está enviando, el otro solo puede recibir y viceversa. La capacidad total del canal es usada por aquel de los dispositivos que está transmitiendo. Los walkie-talkies son un ejemplo de estos sistemas (Forouzane, 2007).

2.3.1.4.3 Full Duplex

También llamado Dúplex, es un modo donde ambas estaciones pueden enviar y recibir simultáneamente. En el modo Full-Dúplex, las señales que van en cualquier dirección deben compartir la capacidad del enlace. Este reparto se puede dar de dos maneras: o bien el enlace tiene que tener caminos de transmisión físicamente separados, uno para enviar y otro para recibir, o es necesario dividir la capacidad del canal entre las señales que viajan en direcciones opuestas. Un ejemplo de este sistema es la red telefónica. Cuando dos personas están hablando por teléfono, ambas pueden hablar y escuchar al mismo tiempo (Forouzane, 2007).

2.3.1.5 Medio o soporte físico

2.3.1.5.1 Guiado o cableado

Medio	Velocidad de transmisión	Inmunidad CEM	Distancia máxima entre equipos	Coste	Ejemplo grafico
Par trenzado (UTP)	Med: 100kbits/s Max.500kbits/s	Muy baja	200m	Bajo	 *Figura 13 Cable UTP*
Par trenzado apantallado (FTP),(STP,(S/STP)	Med: 100kbits/s Max.500kbits/s	Media	1Km	Bajo	 *Figura 14 Cable S/STP*
Coaxial (banda base)	Med: 1Mbits/s Max.50Mbits/s	Media	2,5Km	Medio	 *Figura 15 Cable coaxial para comunicación de datos*
Coaxial (banda ancha)	Med: 300Mbits/s	Media	Entre 10Km y 50 Km	Medio	 *Figura 16 Cable coaxial TV*
Fibra óptica (multimodo)	1Gbps 10Gps	Alta	Entre 2Km y 10 Km	Alto	 *Figura 17 Cable óptico más utilizado*
Fibra óptica (monomodo)	100Gbps 1000Gbps	Alta	Máxima 100Km	Alto	 *Figura 18 Cable óptico usado para exteriores*

Tabla 1 Cables utilizados en las comunicaciones

2.3.1.5.2 No guiado o inalámbrico

Wifi, bluetooth, radiofrecuencia, infrarrojos, microondas.

Figura 20 Icono de señal wireless

Figura 19 Icono bluetooth

2.3.1.5.2 Conectores

Cable UTP	Cable Coaxial	Cable Fibra
RJ-11, RJ-45, RJ-48c, USB, IEEE 1294 (Fireware), DB-9(RS-232), DB-25	BNC, F-Type	MT-RJ, ST, LC, SC, FC

Tabla 2 Tipos de conectores utilizados

2.3.2 Modelos de referencia (arquitecturas)

2.3.2.1 Modelo OSI

Este modelo se basa en una propuesta desarrollada por la Organización Internacional de Normas (ISO) como el primer paso hacia la estandarización internacional de los protocolos utilizados en las diversas capas (Day y Zimmerman, 1983). Este modelo se revisó en 1995 (Day, 1995) y se le llama Modelo de referencia OSI (Interconexión de Sistemas Abiertos, del inglés *Open Systems Interconnection*) de la iso puesto que se ocupa de la conexión de sistemas abiertos; esto es, sistemas que están abiertos a la comunicación con otros sistemas. Para abreviar, lo llamaremos modelo OSI.

Tienes que saber que el modelo OSI en sí no es una arquitectura de red, ya que no especifica los servicios y protocolos exactos que se van a utilizar en cada capa. Sólo

indica lo que una debe hacer. Sin embargo, la ISO también ha elaborado estándares para todas las capas, aunque no son parte del modelo de referencia en sí. Cada uno se publicó como un estándar internacional separado. Aunque el *modelo* (en parte) es muy usado, los protocolos asociados han estado en el olvido desde hace tiempo.

El modelo OSI tiene siete capas.

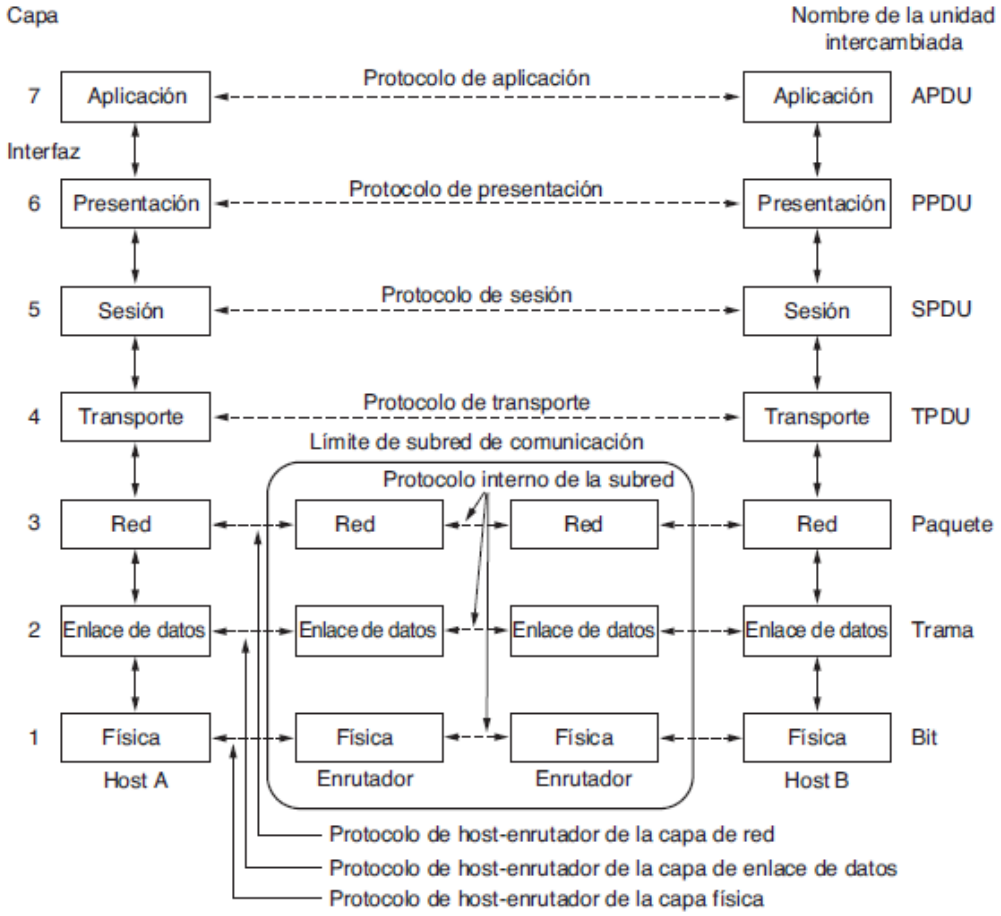


Figura 21 Esquema modelo OSI

La capa física se relaciona con la transmisión de bits puros a través de un canal de transmisión. Los aspectos de diseño tienen que ver con la acción de asegurarse que cuando uno de los lados envíe un bit 1 el otro lado lo reciba como un bit 1, no como un bit 0. En este caso las preguntas típicas son: ¿Que señales eléctricas se deben usar para representar un 1 y un 0?, ¿cuántos nanosegundos dura un bit?, ¿la transmisión puede

proceder de manera simultánea en ambas direcciones?, ¿cómo se establece la conexión inicial y cómo se interrumpe cuando ambos lados han terminado?, ¿cuántos pines tiene el conector de red y para qué sirve cada uno? Los aspectos de diseño tienen que ver con las interfaces mecánica, eléctrica y de temporización, así como con el medio de transmisión físico que se encuentra bajo la capa física.

2.3.2.2 Modelo TCP/IP

Comparación modelo OSI con el modelo TCP/IP

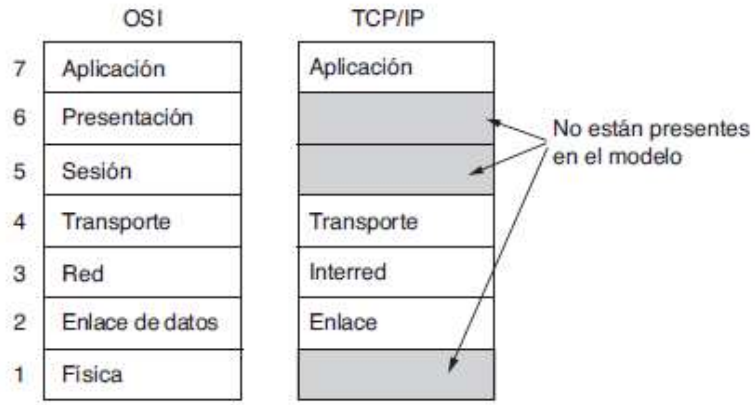


Figura 22 Esquema modelo TCP/IP

Protocolos asociados en cada capa:

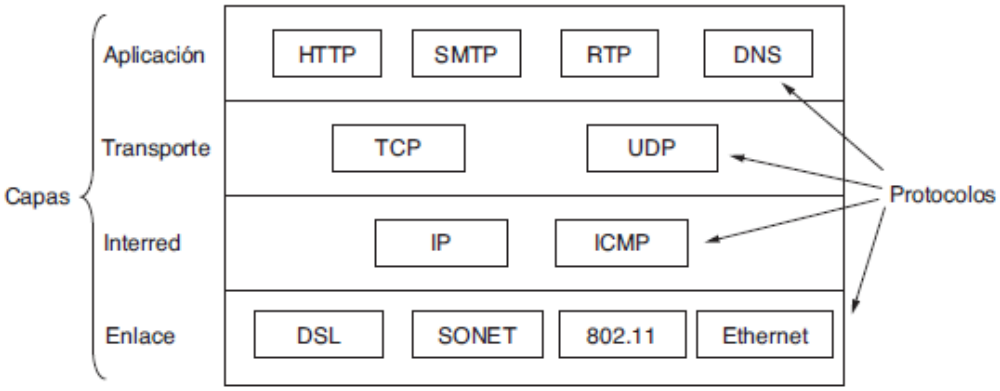


Figura 23 Protocolos de cada capa en el modelo TCP/IP

2.4 Comunicaciones Industriales

2.4.1 Formas

2.4.1.1 Serie (Buses de comunicación)

El emisor es el maestro y los receptores les llamaremos esclavos

Solo el maestro es quien puede iniciar la comunicación.

El maestro dispone de dos modos de comunicación distintos para dirigirse a cada uno de los nodos esclavos:

- Modo unidifusión, el maestro se dirige directamente a un nodo esclavo específico de forma individual. Por lo tanto es necesario que cada esclavo tenga asignada una dirección única.

- Modo difusión, el maestro envía una solicitud a todos los nodos esclavos. Para los mensajes de difusión el maestro utiliza la dirección 0. Todos los nodos esclavos de la red deben aceptar los mensajes de difusión claramente identificables por la dirección 0. Los nodos esclavos tan solo aceptan mensajes de difusión, no los responden.

La imagen de abajo ilustra el proceso y los pasos que sigue una comunicación maestro-esclavo.

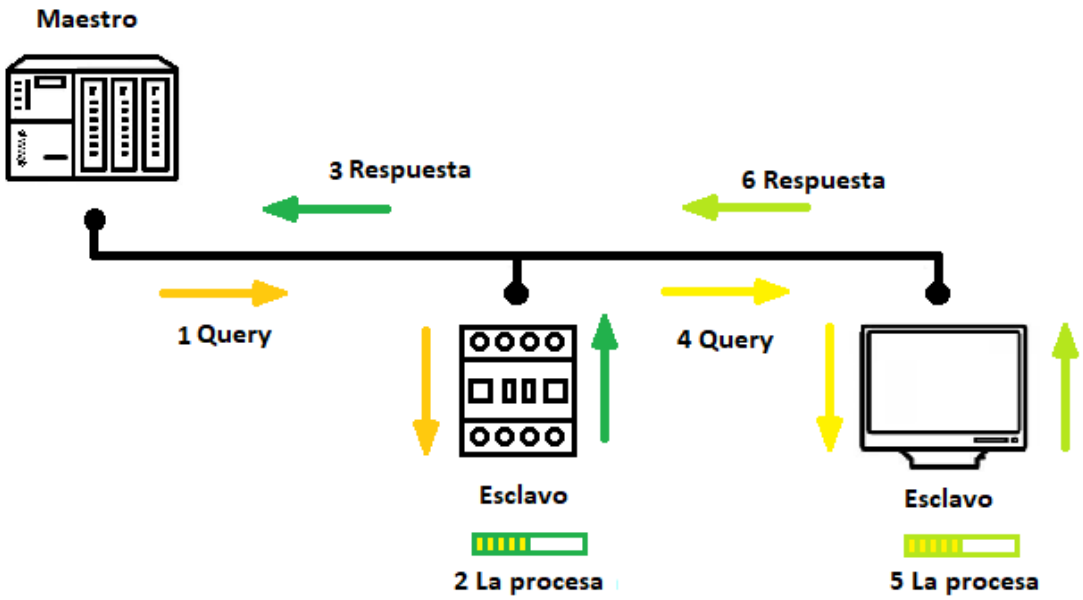


Figura 24 Comunicación típica maestro-esclavo en el bus de campo

1. El maestro inicia la comunicación a un esclavo, envía una petición

2. El esclavo la recibe y la procesa

3. Da una respuesta que llega al maestro y este la procesa

4. Solo entonces puede enviar otra petición.

2.4.1.2 Paralelo (Redes de comunicación)

El emisor es el cliente y los receptores les llamaremos servidores.

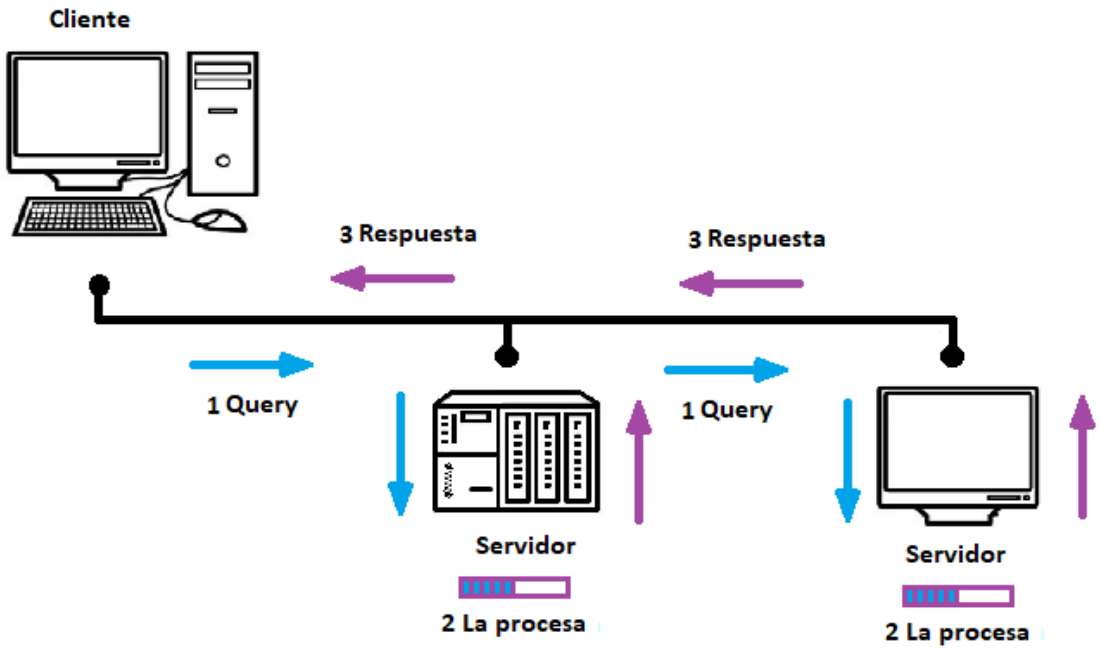


Figura 25 Comunicación cliente-servidor

1. El cliente es capaz de enviar una o múltiples peticiones a los receptores.

2. Los servidores son capaces de recibir múltiples mensajes, procesan y gestionarlas simultáneamente.

3. Es capaz de recibir las respuestas al mismo tiempo

2.4.2 Sistemas de comunicación

Sistema de comunicaciones	Tipología	Media físico	Velocidad de transmisión	Distancia máxima por segmento	Nodos por segmento	Protocolos asociados
PROFIBUS	Bus lineal	Par trenzado apantallado	De 9,6 Kbps hasta 12 Mbps	De 200 m hasta 1200 m	32-64-127	PROFIBUS DP PROFIBUS PA
Modbus	Bus lineal	Par trenzado apantallado	De 300 bps a 19,2 kbps	1000 m	32-64-248	Modbus-RTU
DeviceNet	Bus lineal	Par trenzado apantallado	De 125 kbps hasta 500 kbps	De 200 m hasta 1200 m	64	DeviceNet
CAN	Bus lineal	Par trenzado apantallado	De 50 kbps a 1 Mbps	De 40 m hasta 1000 m	64-127	CANopen
M-Bus	Bus lineal	Cable de 2 hilos	De 300 a 9600 bps	1000 m	250	M-Bus
HART COMMUNICATION FOUNDATION	Bus lineal	Cable de 2 hilos	1200 bps	3000 m	30	HART COMMUNICATION FOUNDATION
AS-i INTERFACE	Bus lineal Árbol Estrella	Cable de 2 hilos	167 kbps	100 m	32	AS-i INTERFACE
INDUSTRIAL ETHERNET	Malla Estrella	Par trenzado apantallado Fibra óptica	De 10 Mbps hasta 100 Mbps De 100 Mbps a 1 Gbps	100 m	n/a	Modbus TCP EtherNet/IP PROFINET

2.5 Comunicación serie (UART)

Existen muchas formas de comunicación serie, pero como se envían los datos en una comunicación serie esta estandarizado a través de su puerto serie. Por tanto todos los dispositivos que usan el puerto serie funcionan exactamente igual.

El hardware encargado de enviar los datos a través del puerto serie es el UART (Universal Asynchronous Reciver Transmiter).

Si hablamos de un microcontrolador, la UART está integrada dentro del micro.

En caso de usar un FPGA, tendríamos que diseñar nuestra propia UART.

La UART se encarga de gestionar las comunicaciones, enviar los datos a través del puerto serie, de gestionar las interrupciones de entrada, si fuera necesario, y de almacenar los datos recibidos de manera temporal.

En función de la cantidad de bits que se transmiten a la vez la comunicación puede ser con puerto serie o puerto paralelo.

Con un puerto serie se transmite un bit de información por vez en el tiempo, así que toma más tiempo la transmisión de datos.

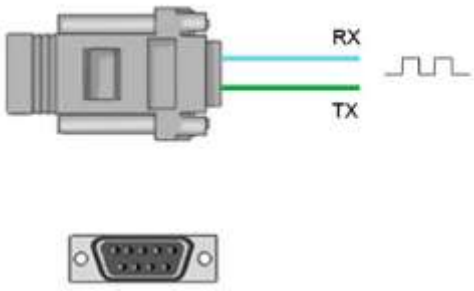


Figura 26 Conector DB9

A diferencia del puerto serie el puerto paralelo se transmiten 2 o más bits (hasta 8) de información en simultaneidad. Para conseguir esto como contra se requerirán más pines en el conector.

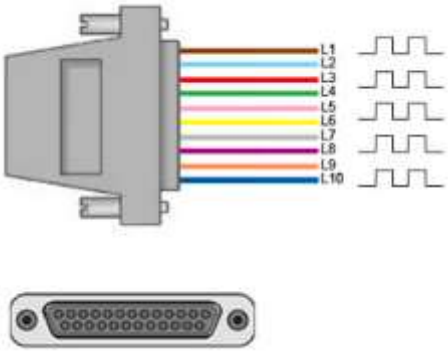


Figura 27 Conector DB-25

Bits y trama

Por ejemplo si queremos enviar el carácter "a".

Carácter	Tabla ascii	binario
a	97	01100001

Tabla 3 Conversión carácter "a"

Para enviarlo deberemos incluirlo dentro de una trama.

1. Invertiremos el orden de los bits ya que se envía del bit menos significativo al más significativo.

2. La línea de transmisión siempre que está en reposo estará en nivel alto.

3. Para iniciar la comunicación debemos enviar un bit de "start" que en este caso es 0 y lo mantendremos durante un tiempo que llamaremos tiempo de bit.

4. Pasado el tiempo de bit empezaremos la a enviar los datos.

5. Cada bit lo mantendremos en la línea de datos durante el tiempo de bit que hemos definido en el punto 3.

6. Una vez enviamos los datos tenemos que liberar la línea e indicar que ya hemos transmitido, para ello enviaremos un bit de stop enviaremos un 1 durante el tiempo de bit.

7. Tras enviar el bit de stop la línea quedara en espera, nivel alto

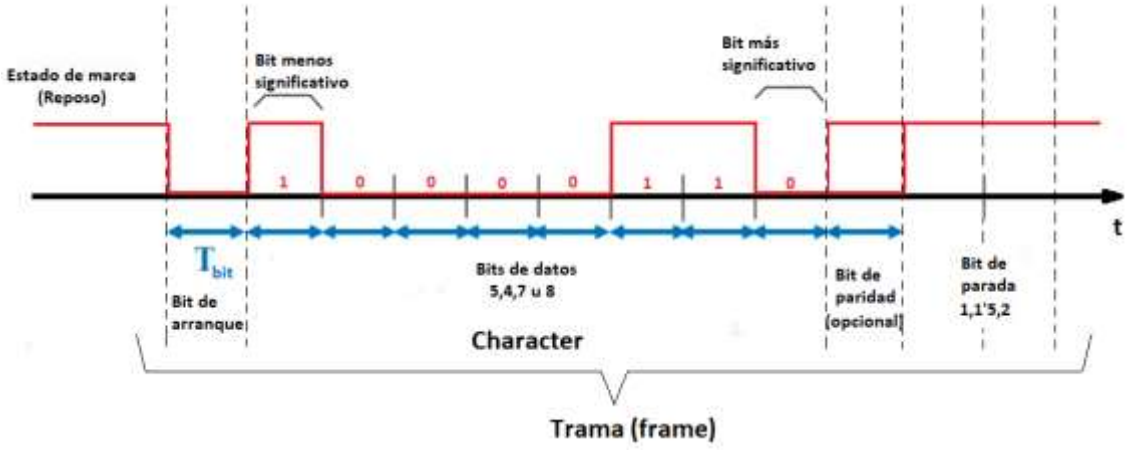


Figura 28 Trama de datos interna de un paquete

A día de hoy no suele utilizarse el bit de paridad pero si se usara seria antes del bit de parada.

La Tbit, es el tiempo que mantendremos cada bit en la línea, bien sea un cero o un uno.

Ese tiempo de bit ira condicionado por la tasa de baudios (baud rate). Generalmente la tasa de baudios es de 9600, esto indica cuantos bits podemos enviar en un segundo y lógicamente antes de que empiece la comunicación ambos dispositivos tienen y deberían poderse configurar igual.

El tiempo de bit se puede calcular como:

$$Tiempo\ de\ bit = \frac{1}{Baud\ rate}$$

En el caso ejemplo,

$$T_{Bit} = \frac{1}{9600} = 104{,}2\mu s$$

En conclusión si queremos enviar en el caso ejemplo la letra "a" a 9600 baudios, debemos mantener el bit de arranque durante 104,2μs y los restantes se enviaran también a ese tiempo.

Por ultimo saber que en las comunicaciones digitales lo que vamos a tener es una línea de masa, y con respecto a esa línea de masa vamos a tener tensiones. Si hablamos de capa física en TTL diremos que los 0 se conviertan en 0 voltios y los 1 en 5 o 3,3 voltios.

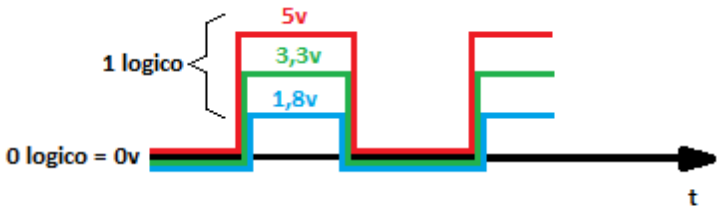


Figura 29 Envió de datos TTL sin puerto

2.5.1 Estándares físicos (puertos)

Estos estándares implementan la trama de comunicación serie asíncrona (UART).

Quiere decir que utilizan las mismas ideas con respecto a la trama, la manipulación de datos y los tiempos de bit, mencionado en al apartado anterior sin embargo varían en la forma de llevar los bits al mundo físico.

En el caso de los puertos RS232 y RS485 la característica principal es que son diferencial. Sus tensiones no están referenciadas a masa. Se unen a través de dos cables que van a tener siempre tensión superior a la de masa y por tanto nunca van a estar a masa. Lo que nos interesa no es la tensión de los cables con respecto a masa si no la diferencia de tensione en estos dos cables. De esta manera cuando una interferencia electromagnética afecta a nuestra base de comunicación RS232 o RS485 afectara a los dos cables por igual, conservando la misma diferencia de tensión que lo original (sin interferencia electromagnética).

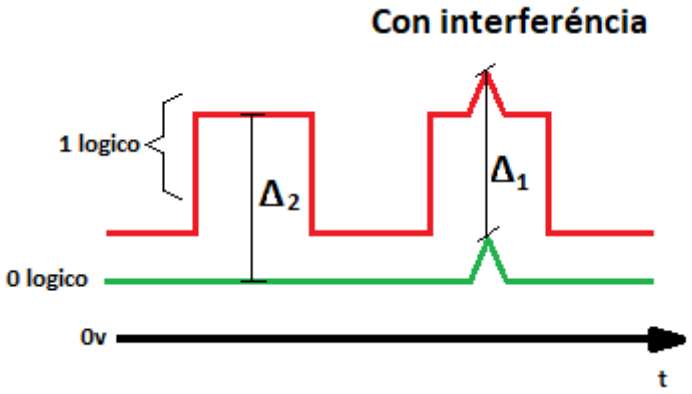


Figura 30 Envió de datos utilizando puerto

2.5.1.1 RS 232

Como hemos dicho es una interfaz que amplía la señal a los niveles de tensión y corriente requeridos. Adecuado para velocidades bajas y de corta distancia. La distancia del cable como máximo puede ser de 15 metros.

El modo de operación a utilizar puede ser Simplex o Full dúplex.

Permite un dispositivo transmisor y uno receptor en una misma línea de bus.

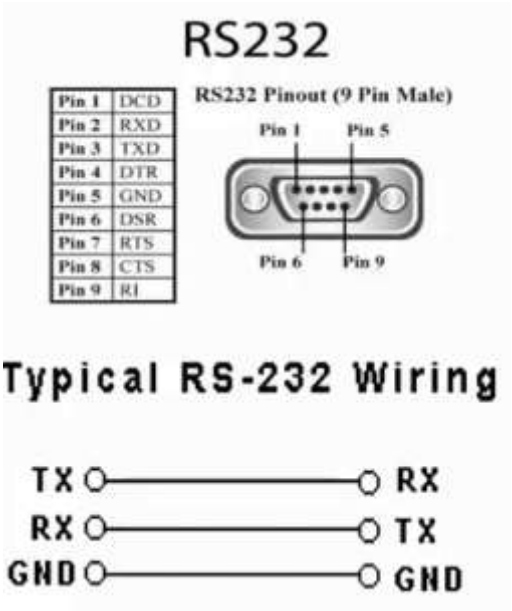


Figura 31 Tabla pines y conexión del RS232

2.5.1.2 RS 485

Con respecto al RS232 en el RS485 cambiarían el amplificador y el receptor. Adecuado para velocidades más altas en distancias largas. La distancia del cable puede ser de 1Km.

El modo de operación a utilizar puede ser Simplex o Half-Duplex

Permite comunicar hasta 32 dispositivos en una misma línea de bus.

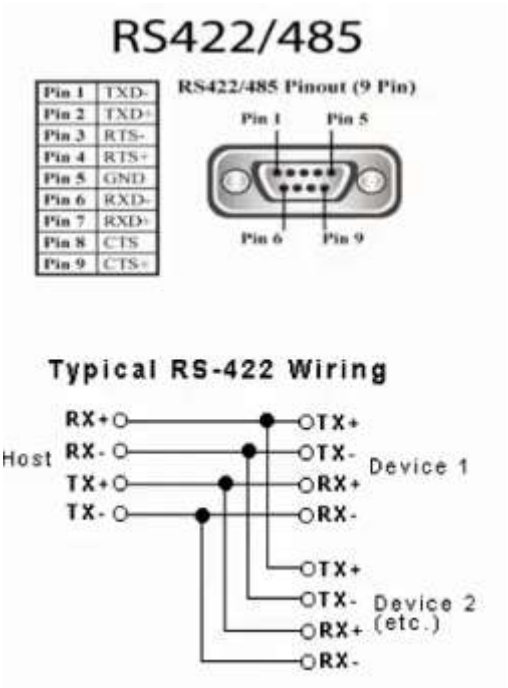


Figura 32 Tabla pines y conexión del RS485

2.6 Comunicación Ethernet

Ethernet es la evolución de la comunicación serie. Es una tecnología ampliamente utilizada en redes de área local bajo el estándar internacional IEEE 802.3 como base. Hace uso del protocolo CSMA/CD para evitar colisiones tecnológicas.

Ethernet funciona con la técnica de modulación de Manchester útil en la transferencia de datos binarios en base a señales analógicas, RF, óptico, señales digitales de alta velocidad digital o de larga distancia. Es un método de codificación eléctrica de una señal binaria en el que en cada tiempo de bit hay una transición entre dos niveles de señal. Es una codificación autosincronizada, ya que en cada bit se puede obtener la señal de reloj, lo que hace posible una sincronización precisa del flujo de datos. Una desventaja es que consume el doble de ancho de banda que una transmisión asíncrona. Hoy en día hay numerosas codificaciones (8b/10b) que logran el mismo resultado pero consumiendo menor ancho de banda que la codificación Manchester.

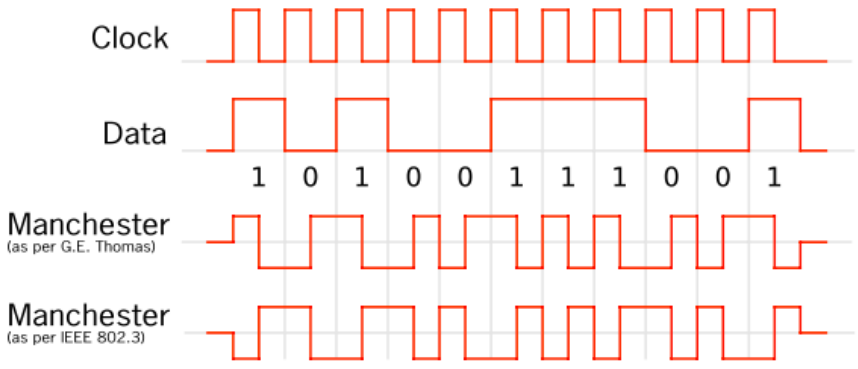


Figura 33 Codificación Manchester

Trama de Ethernet:

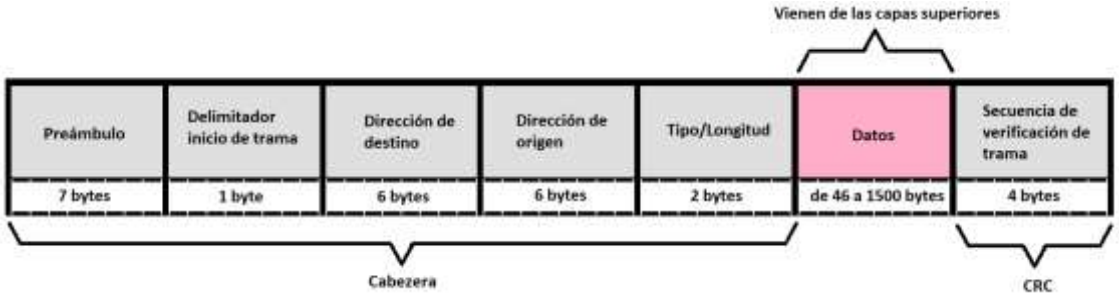


Figura 34 Trama Ethernet

Preámbulo: Patrón de unos y ceros que indica a las estaciones receptoras que una trama es Ethernet (IEEE 802.3). La trama Ethernet incluye un byte adicional que es el equivalente al campo Inicio de Trama (SOF) de la trama IEEE 802.3.

Inicio de trama (SOF): Byte delimitador de IEEE 802.3 que finaliza con dos bits 1 consecutivos, y que sirve para sincronizar las porciones de recepción de trama de todas las estaciones de la red. Este campo se especifica explícitamente en Ethernet.

Direcciones destino y origen: Incluye las direcciones físicas (MAC) únicas de la máquina que envía la trama y de la máquina destino. La dirección origen siempre es una dirección única, mientras que la de destino puede ser unicast (trama enviada a una sola máquina), de multicast (trama enviada a un grupo) o de broadcast (trama enviada a todos los nodos).

Tipo (Ethernet): Especifica el protocolo de capa superior que recibe los datos una vez que se ha completado el procesamiento Ethernet. **Longitud (IEEE 802.3):** Indica la cantidad de bytes de datos que sigue este campo.

Datos: Incluye los datos enviados en la trama. En las especificación IEEE 802.3, si los datos no son suficientes para completar una trama mínima de 64 bytes, se insertan bytes de relleno hasta completar ese tamaño (tamaño mínimo de

trama). Por su parte, las especificaciones Ethernet versión 2 no especifican ningún relleno, Ethernet espera por lo menos 46 bytes de datos.

Secuencia de verificación de trama (FCS): Contiene un valor de verificación CRC (Control de Redundancia Cíclica) de 4 bytes, creado por el dispositivo emisor y recalculado por el dispositivo receptor para verificar la existencia de tramas dañadas. Cuando un paquete es recibido por el destinatario adecuado, les retira la cabecera de Ethernet y el checksum de verificación de la trama, comprueba que los datos corresponden a un mensaje IP y entonces lo pasa a dicho protocolo para que lo procese. El tamaño máximo de los paquetes en las redes Ethernet es de 1500 bytes.

2.7 Protocolo Modbus

Modbus permite el control de una red de dispositivos de forma rápida y sencilla. Dispositivos de diferente naturaleza implementan este protocolo: controles electrónicos, variadores de velocidad, ordenadores,…

Cada dispositivo debe tener un número asignado para ser identificado en la red.

Hay tres implementaciones básicas para Modbus:

Dos para transmisión de datos a lo largo de una interfaz serie, modem EIA/TIA-232-E (RS232), EIA-422, EIA/TIA-485-A (RS-485), óptica y conexión de radio:

- Modbus RTU
- Modbus ASCII

Y para transmisión de datos a través de Ethernet.

- Modbus TCP/IP

Trama general Modbus:

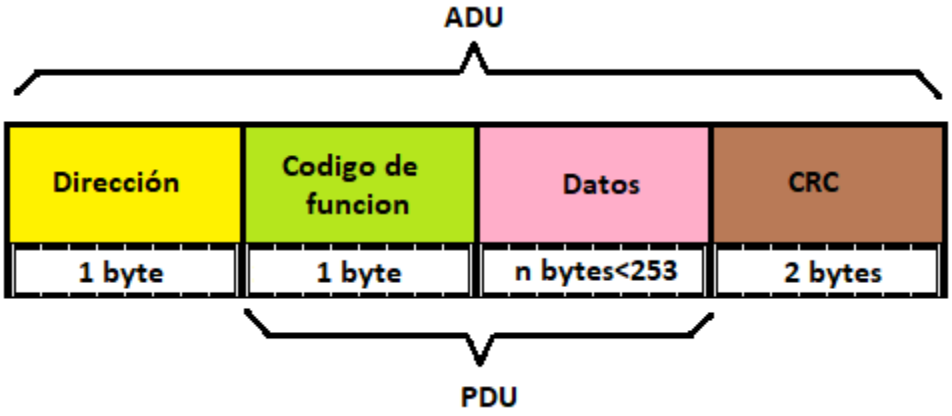


Figura 35 Trama modelo Modbus

ADU: (Application Data Unit)

PDU: (Protocol Data Unit)

2.7.1 RTU

Protocolo que permite el control y la supervisión a través del bus de campo.

La trama de Modbus RTU sigue esta estructura:

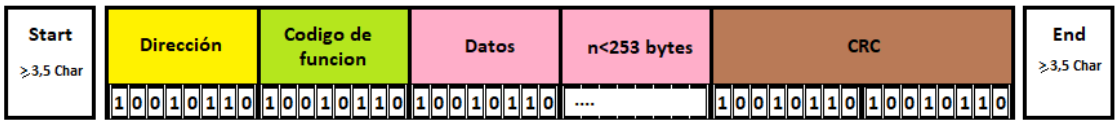


Figura 36 Trama Modbus RTU

Dirección: es la dirección del dispositivo esclavo al que se dirige al registro. Cada dispositivo corresponde a una sola dirección. Del mismo modo, la respuesta comienza con la dirección del dispositivo esclavo. Las direcciones de los dispositivos son del 1 al 247. La dirección 0 se usa para el direccionamiento de paquetes y es reconocida por todos los dispositivos: las direcciones en el rango 248-255 están reservadas.

Código de función: Se asigna el comando de ejecución.

Datos: Este campo contiene información que es necesaria para que el dispositivo ejecute el comando dado por el maestro, o por el contrario contiene la información requerida cuando se devuelve una respuesta al maestro. La longitud del formato depende del número de la función y varía en el rango de 0-252.

CRC: Control de suma para verificación de errores en la trama. Durante la transmisión de línea, el byte menos significativo de la suma de control se escribe primero.

2.7.2 TCP/IP

Protocolo que permite el control y la supervisión a través de internet.

La trama de Modbus RTU sigue esta estructura:

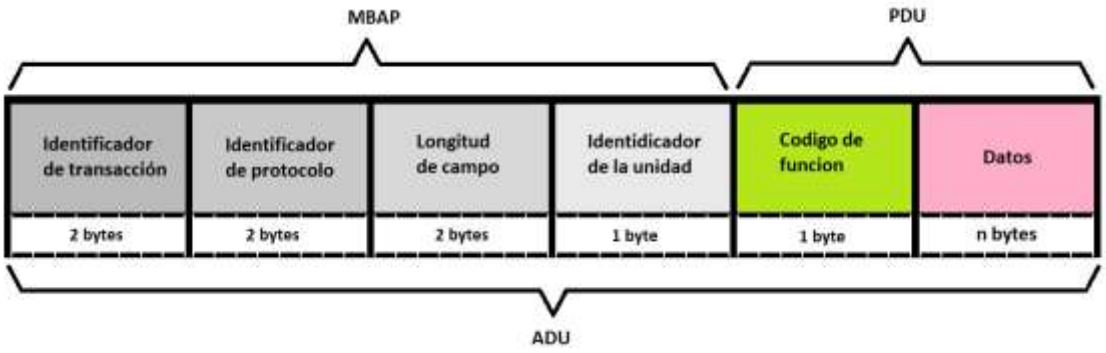


Figura 37 Trama Modbus TCP/IP

Modbus Application Protocol (MBAP) header

ID transacción: Este campo de identificación se utiliza para el emparejamiento de transacciones cuando un cliente envía múltiples mensajes a lo largo de la misma conexión TCP sin esperar una respuesta previa.

ID protocolo: Este campo siempre es 0 para servicios Modbus y otros valores están reservados para futuras extensiones.

Longitud: Este campo es un recuento de bytes de los campos restantes e incluye el byte identificador de la unidad, el byte del código de función y los campos de datos.

ID unidad: Este campo se usa para identificar un servidor remoto ubicado en una red que no sea TCP / IP (para puente en serie). En una aplicación típica de servidor Modbus TCP / IP, el ID de la unidad se establece en 00 o FF, el servidor lo ignora y simplemente repite la respuesta.

Modbus TCP / IP utiliza "TCP / IP" y "Ethernet" para transportar los datos de la estructura de mensajes Modbus entre dispositivos. Es decir, Modbus TCP / IP combina una red física (Ethernet), con un estándar de red (TCP / IP) y un método estándar de representación de datos (Modbus).

La figura 38 ilustra la construcción de un paquete TCP / IP-Ethernet para la transmisión. Para Modbus TCP / IP, la capa de aplicación es Modbus y la Unidad de datos de aplicación Modbus está integrada en la matriz de datos TCP. Cuando una aplicación envía sus datos a través de la red, los datos pasan a través de cada capa; observe cómo la información de la capa superior se envuelve en los bytes de datos de la siguiente capa más baja (encapsulada). Cada capa posterior tiene una función designada y adjunta su propio encabezado de protocolo al frente de su paquete. La capa más baja es responsable de enviar los datos. Todo este procedimiento de envoltura se revierte para los datos recibidos (los datos recibidos se desenvuelven en cada nivel y se pasan a la capa de aplicación del receptor).

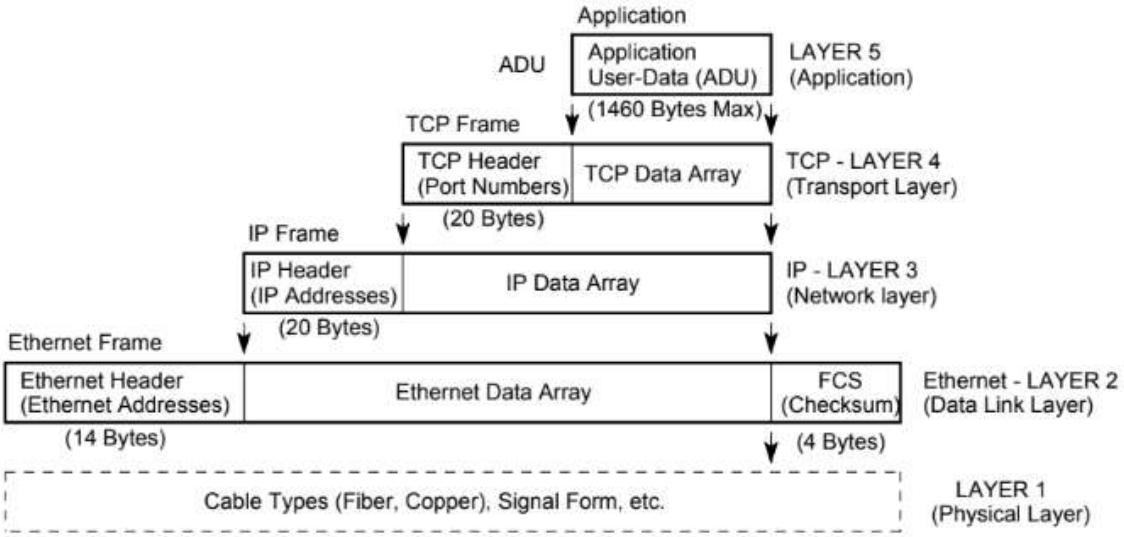


Figura 38 Modelo de capas Modbus TCP/IP

2.8 TCP vs UDP

TCP (Transmission Control Protocol) Es un protocolo que garantiza la entrega de datos.

Se usa para enviar mensajes por Internet de un dispositivo a otro por medio de conexiones virtuales, siendo útil para aplicaciones que requieren confiabilidad alta y donde el tiempo de transmisión es menos crítico.

TCP hace control de flujo y maneja confiabilidad y control de congestión. Eso quiere decir que requiere de tres paquetes para establecer una conexión antes de transmitir.

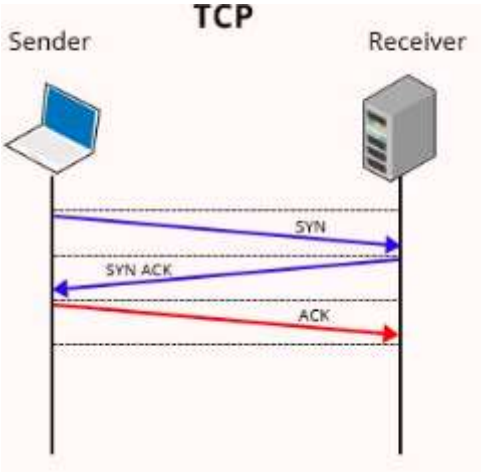


Figura 39 Protocolo de comunicación TCP antes del envió de datos de información

Algunos ejemplos de protocolos de aplicación que usan TCP son: HTTP, HTTPs, SMTP, FTP, Telnet.

UDP (User Datagram Protocol) No garantiza la entrega de datos, no le importa si el otro dispositivo recibe los datos. Se usa para transporte de mensajes y/o transferencias sin conexiones (significa que un programa puede enviar una carga de paquetes de data y hasta ahí llega esa relación), siendo útil para aplicaciones que necesitan transmisión rápida y efectiva como los servidores que reciben una gran cantidad de peticiones pequeñas de un alto número de clientes.

UPD no tiene control de flujo.

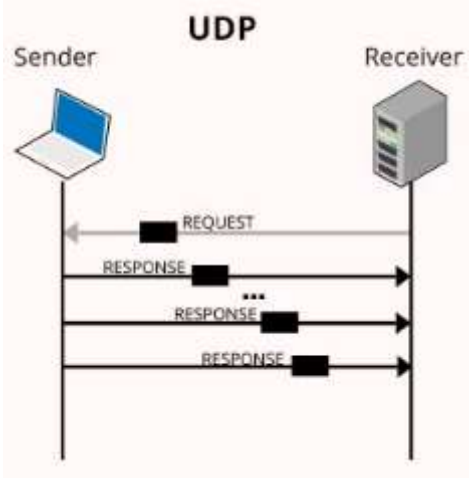


Figura 40 Comunicación UDP

Algunos ejemplos de protocolos de aplicación que usan TCP son: DNS, VoIP, TFTP, DHCP, SNMP, RIP.

2.9 Analizador de redes

Es un instrumento capaz de analizar las propiedades de las redes eléctricas, especialmente los parámetros de dispersión (propiedades asociadas con la reflexión y la transmisión de señales eléctricas) .Los analizadores de redes son más frecuentemente usados en altas frecuencias, las frecuencias de operación pueden variar de 5Hz a 1,05THz.

Para más información ver el manual del analizador utilizado en el Anexo IV.

2.10 Pasarelas

Figura 41 Ejemplo pasarela

Una pasarela es un equipo de interconexión previamente configurado que permite el flujo de comunicación entre dos protocolos distintos.

Normalmente los protocolos de comunicaciones utilizados en la industria de manera local son serie y para su uso y tratado de la señal fuera de la celda es necesario convertir y procesar esos datos mediante otro tipo de protocolo utilizado con cable Ethernet, por ejemplo, para comunicarse remotamente a distancia.

En este caso se instala una pasarela que realiza la conversión RS-485 (Modbus) a Ethernet (TCP/IP) de la que pueden colgar hasta 32 equipos a una velocidad de 115000 bits por segundo.

Para más información de la pasarela utilizada ver anexo V.

Para la comunicación entre analizador y la pasarela, ver las configuraciones hechas por Pumares Cerezo, J. en el anexo VI.

2.11 Software de red

2.11.1 NodeRed

La programación visual basada en diagramas de flujo fue inventada por J.Paul Morrison el año 1970.

NodeRed es un software desarrollado originalmente por IBM para conectar dispositivos de hardware, API y servicios en línea como parte de IoT.

En 2013 se convirtió en un programa "open-source" y esta mantenido por la organización JS Foundation desde 2016.

JS Foundation lo definen como:

"Aplicación para programar visualmente flujos de información que permiten la conectividad entre dispositivos, APIs y servicios online." (JS Foundation, 2019)

Básicamente nos viene a decir que es una herramienta que facilita la interconexión entre dispositivos físicos como PLC, IoT,.. y/o virtuales.

2.11.2 InfluxDB

Es un gestor base de datos de series temporales, "Time Series Database", que guarda series de datos indexadas a lo largo del tiempo.

Programado en lenguaje "Go" permite la interacción vía API HTTP(S) (JSON) e interfaz web. Los datos se gestionan con un lenguaje similar al SQL.

Es utilizado comúnmente como solución de almacenamiento donde involucre el uso de grandes cantidades de datos situadas en instantes de tiempo diferentes.

Gracias a esta herramienta "open-source" podremos crear un servidor de manera local u online, donde guardar dichos los datos.

Otros gestores de base de datos son:

- MySQL
- Microsoft SQL Server
- Oracle Database
- SQLite

2.11.3 Grafana

Es una herramienta de visualización de series de datos temporales. Muy utilizada para visualizar datos en tiempo real de infraestructuras y aplicaciones o en servicios de monitorización.

Grafana está escrito en lenguaje “Go” y tiene un API HTTP.

En dicho programa “open-source” podremos crear un entorno de visualización mediante la creación de “dashboards” haciendo uso de un abanico de alternativas para personalizar la apariencia de los datos leídos. También es capaz de integrar notificaciones de alertas a tiempo real, así como aplicar lógica sobre la información leída.

CAPÍTULO 3. DISEÑO

3.1 Arquitectura del sistema

Para entender el flujo de datos representaremos nuestro caso estudio en base a los elementos de intervención de la celda estudio y a los programas "open source" utilizados. Su jerarquía es la que se muestra en la imagen inferior.

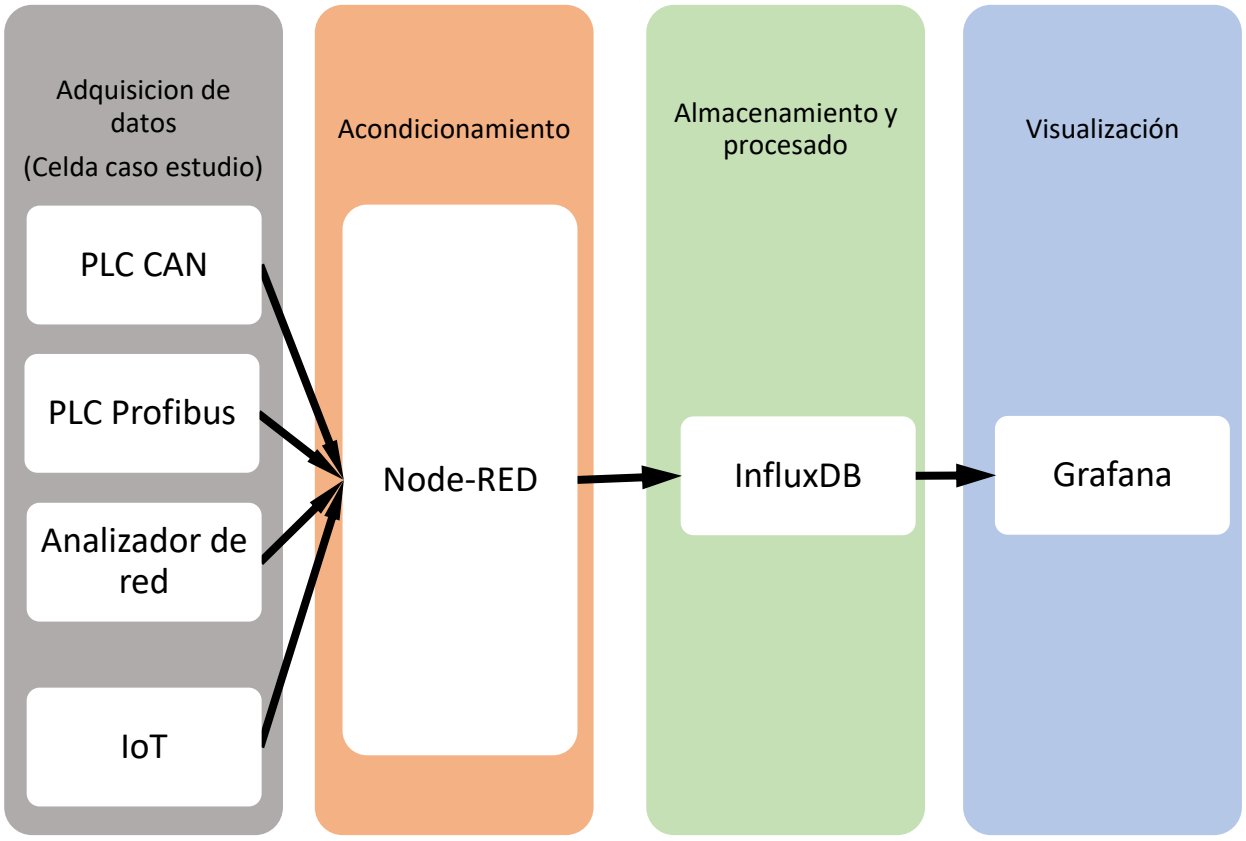


Figura 42 Diagrama arquitectura caso estudio

3.2 Hardware del sistema

Nivel de campo

En la imagen de abajo podemos ver representados los dispositivos que intervienen (sensores, actuadores, motores, etc,..) y como están distribuidos en la celda de producción.

Cada uno de ellos irá conectado a su periférico numerado y ubicado fuera de la celda.

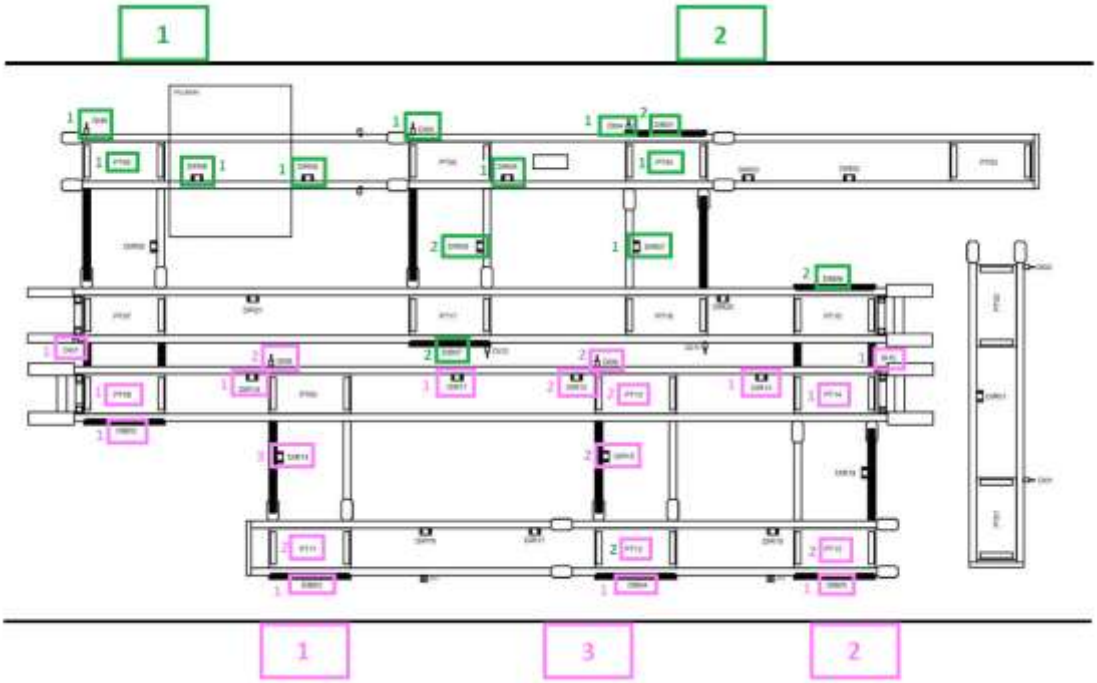


Figura 43 Celda caso estudio sacada del Anexo I

Los periféricos irán conectados vía bus de campo a su PLC correspondiente utilizando el protocolo requerido. Los periféricos 1, 2 y 3 de color lila irán conectados mediante cable de par trenzado apantallado al PLC utilizando el protocolo Profibus gracias al puerto RS485.

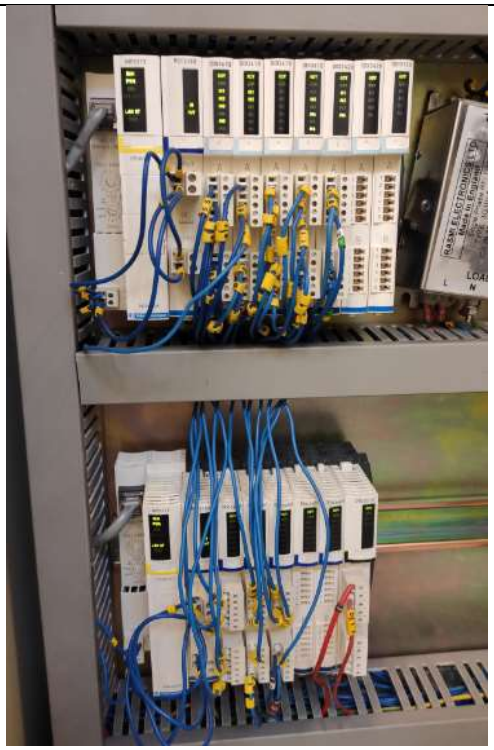
Los periféricos 1 y 2 de color verde irán conectados mediante cable CAN al PLC utilizando el protocolo CanOpen gracias al conector DB9.

Otro caso muy distinto son los periféricos del pulmón que están conectados mediante cable de Ethernet, UTP, utilizando el protocolo Modbus TCP/IP que será el mismo utilizado aguas arriba.

3.2.1 Equipos de campo

	Foto	Descripción	Protocolo asociado
PROFIBUS		Periférico 1 (Isla Advantys STB)	Profibus RS485
		Periférico 2 (Isla Advantys STB)	
		Periférico 3 (Isla Advantys STB)	

		PLC Profibus (Modicon TSX P57)	
C A N		Periférico 1 (Isla Advantys STB)	CAN
		Periférico 2 (Isla Advantys STB)	
		PLC CAN (Modicon M340)	

ETHERNET		Periféricos (Pulmón) (Isla Advantys STB)	Modbus TCP/IP
		PLC Pulmon (Modicon M340)	
		Analizador de redes CVM-MINI-MC	Modbus RTU RS485

		Pasarela TCP1RS+	Modbus RTU RS485

Tabla 4 Equipos involucrados en el fieldbus, tipo de conexión y el protocolo ce comunicación utilizado

3.2.2 Equipos de Red

Nivel de red

En este nivel los PLC y el analizador de red (pasarela) y todos los dispositivos más allá, utilizaran cable de Ethernet (UTP), para conectarse al switch, router y con ello a la WAN. El protocolo utilizado para el envío y recepción de datos será el Modbus TCP/IP.

Para ello nos ayudaremos de la aplicación NodeRed ubicada en nuestro ordenador.

	Foto	Descripción	Protocolo asociado
		PLC Profibus (Modicon TSX P57)	

ETHERNET		PLC CAN (Modicon M340)	Modbus TCP/IP
		PLC Pulmón (Modicon M340)	
		Pasarela TCP1RS+	
		Magelis Edge Box	

Tabla 5 Equipos involucrados a nivel de red, tipo de conexión y el protocolo utilizado entre ellos

3.2.3 Visión general

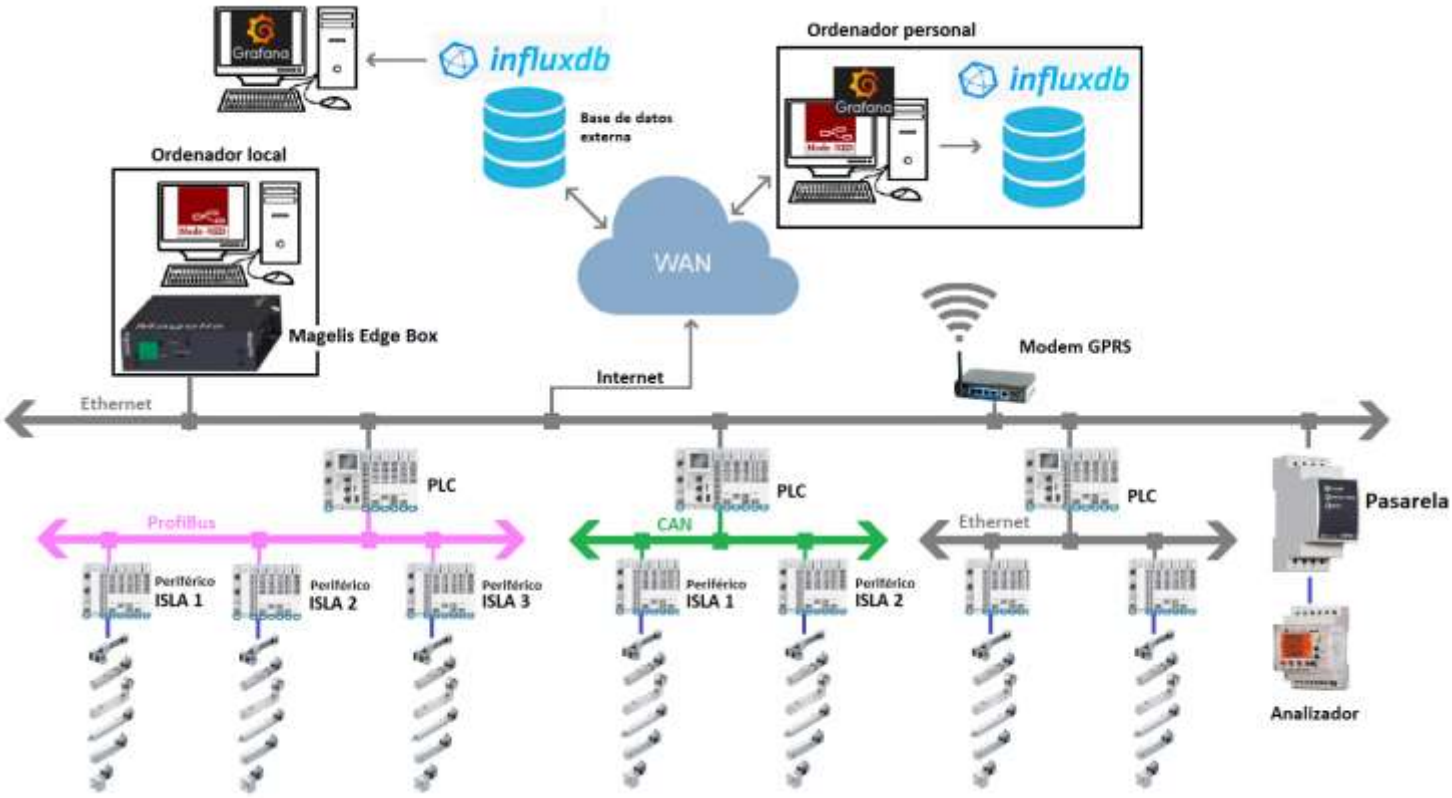


Figura 44 Esquema comunicación hardware-software caso estudio

En la figura 44 podemos ver en todo su conjunto como están conectados íntegramente los elementos que intervienen (referidos en el apartado anterior) y la estructura diseñada para facilitar el flujo de los datos.

En el nivel inferior, tenemos lo que llamamos elementos o dispositivos de campo, en nuestro caso está compuesto por todos los sensores, actuadores, retenedores, analizador de redes, etc., y los módulos de E/S donde van conectados. Estos periféricos son las islas 1,2 y 3 de Profibus, las islas 1 y 2 de CAN y las islas del pulmón.

En un nivel medio encontraremos los equipos de control, compuestos por los PLCs de Profibus (Modicon TSX), CAN (Modicon M340) y el del pulmón (Modicon 340). Todos los ellos irán conectados a sus respectivos periféricos del nivel inferior.

El PLC Modicon TSX ira conectado mediante bus de campo con sus periféricos utilizando el protocolo Profibus. Este protocolo es el llamada maestro-esclavo y tendrá un control distribuido a tiempo real. Este protocolo funciona parecido al de Modbus RTU teniendo una trama un poco más compleja.

Por otro lado tendremos el PLC de CAN que ira conectado a sus periféricos utilizando el protocolo de CAN. Este protocolo funciona por prioridades y es diferente y más complejo que el de Profibus.

El PLC Modicon M340 del pulmón, por lo contrario, utilizará cable Ethernet por lo que empleara el protocolo Modbus TCP/IP, el mismo utilizado aguas arriba.

En un nivel superior encontraremos la red global (WAN), donde los tres PLCs y la pasarela del analizador irán conectados a ella vía Ethernet utilizando el protocolo Modbus TCP/IP.

En este nivel de comunicación de tiempo no crítico encontramos el ordenador local, que está físicamente ubicado dentro del laboratorio. Este ordenador tiene instalado el NodeRed y trabaja paralelamente con la Magelis Edge Box de Schneider.

Para enlazar los PLCs y la pasarela con nuestro ordenador personal nos conectaremos remotamente vía Wifi a la red de la UPC, teniendo acceso a este nivel superior.

Dentro de nuestro ordenador personal tendremos instalado los tres programas necesarios para la obtención y transmisión de datos, y posterior almacenamiento y presentación de estos. Nuestro propio ordenador funcionara como servidor cuando almacenemos los datos, quedando guardados internamente en el ordenador personal. Este proceso se ha utilizado para facilitar la programación y poder trabajar des de casa.

La idea es que una vez el programa este operativo, este se ejecute dentro del ordenador local del laboratorio, direccionando los datos a una IP de un servidor externo que se encuentra las 24h en funcionamiento. Al igual que el servidor externo nuestro ordenador local se encontraría funcionando siempre que la celda estudio esté en funcionamiento.

Esta base de datos externa (Cloud) se ha representado también en la figura 44 pudiendo acceder a ella mediante otro monitor, representado con el icono del Grafana.

En los siguientes capítulos se explicará con más detalle el procesamiento de estos datos efectuados en el nivel superior.

CAPÍTULO 4. PROGRAMACIÓN

4.1 Software del sistema

Como hemos visto antes en la arquitectura del sistema hacíamos uso de los programas "open source" NodeRed, InfluxDB y Grafana para el trato de las señales recibidas de la celda caso estudio.

4.1.1 Adquisición (NodeRed)

NodeRed se ejecuta des de cualquier ordenador, ya sea en el trabajo (un ordenador que se encuentra físicamente en el laboratorio y conectado mediante cable Ethernet a los dispositivos) o personal (remotamente a distancia donde la conexión se realizaría mediante Wifi conectándose a la red local del laboratorio a través del "router".

En el caso estudio se ha trabajado des de un ordenador personal por tanto ejecutaremos NodeRed en él y nos conectaremos a la celda del laboratorio a distancia mediante conexión Wifi.

Para ejecutar el programa se hará uso del terminal cmd del ordenador escribiendo el comando "node-red".

El acceso a NodeRed se realiza mediante una dirección IP y un puerto en la pantalla del navegador de internet. Esta interface se ejecuta utilizando las bibliotecas de software escritas en JavaScript llamadas Node.js. Los datos creados están codificados en el formato abierto JSON.

Normalmente si usas el navegador en el mismo ordenador que está ejecutando NodeRed puedes acceder a él utilizando la URl:

http://127.0.0.1:1880 o que es lo mismo http://localhost:1880

Por el contrario si estas utilizando un navegador en otro ordenador, necesitaras la dirección IP del ordenador que está ejecutando NodeRed:

http://<direccion-IP>:1880

Independientemente de la dirección IP, el puerto que nos viene por defecto es el 1880 que se puede cambiar des de la configuración.

Antes de empezar con la creación de los flujos, obtención de señales y acondicionamiento de datos fue necesario descargar nuevos nodos de la paleta que no venían descargados de origen.

4.1.1.1 Librería Modbus/TCP

La paleta "node-red-contrib-modbustcp" se compone de dos nodos, una de lectura y otro de escritura:

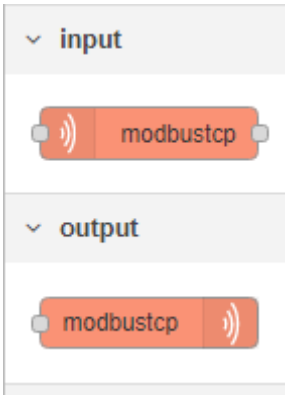


Figura 45 Nodos de lectura y escritura modbustcp

Dichos bloques permiten acceder a los registros de los PLC y analizador de redes diferenciando entre bits y words.

4.1.1.2 Librería InfluxDB

La paleta "node-red-contrib-influxdb" se compone de tres nodos, dos de lectura y uno de escritura.

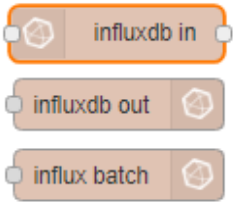


Figura 46 Nodos de lectura y escritura influxDB

Dichos bloques mandan peticiones de escritura al cliente InfluxDB.

4.1.1.3 Creación del flujo

Flujo PLC Profibus y CAN:

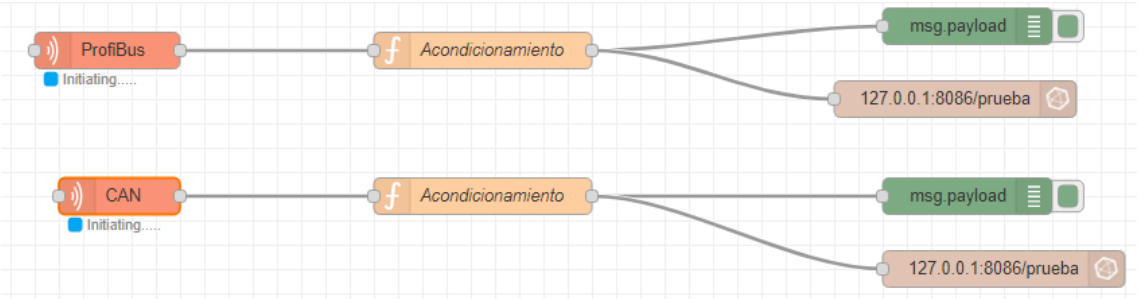


Figura 47 Flujo de adquisición y posterior envió de datos de los PLCs

Utilizamos los nodos "modbustcp" llamados "ProfiBus" y "CAN" de lectura para leer los registros de los PLCs. Seguidamente necesitaremos procesar y acondicionar estos datos en una forma organizada haciendo uso de una arquitectura entendible para la base de datos, InfluxDB. Para crear esta programación utilizaremos los nodos de función que llamaremos "Acondicionamiento".

Como hemos dicho el programa interno del nodo de función dependerá de que tipo de información relevante queramos enviar y la estructura final que debería obtener para ser enviada.

Esta segunda, vendrá condicionada a la utilización del nodo de escritura "influx batch", donde se deben representara los valores a enviar dentro de unos campos llamados "fields", que podremos filtrar-los mediante "tags", asignarles un "timestamp" y estos tres deberán estar dentro del "measurement" con el nombre a escoger usado en nuestra base de datos.

En nuestro caso no haremos uso de los campos,"tags" (para filtrar y agrupar una búsqueda dentro de la base de datos) como tampoco del "timestamp". Nos va bien que coja la hora por defecto de nuestro ordenador donde tenemos ejecutado el programa.

Finalmente esta información acondicionada sería enviada mediante el nodo de escritura "Influx batch" al sevidor de InfluxDB.

El nodo "debug" lo utilizamos para asegurarnos que la información es bien interpretada por el programa y si hay algún problema es debido al bloque de escritura. Por tanto lo que recibe el bloque "debug" tiene que ir en concordancia con los que leemos en la base de datos.

En la siguiente tabla se especifica la ubicación de cada posición de memoria interna del PLC asignada a cada dispositivo de la celda estudio junto con su almacenamiento final.

PLC Profibus				
Dirección	Posición (bit)	Consigna	Programa InfluxDB	Campo InfluxDB
%MW210	%M210.15		15	Reserva
	%M210.14	Avance	14	DIR14
	%M210.13	Petición	13	
	%M210.12	Reposo	12	
	%M210.11	Avance	11	DIR13
	%M210.10	Petición	10	
	%M210.9	Reposo	9	

	%M210.8	Avance	8	DIR12
	%M210.7	Petición	7	
	%M210.6	Reposo	6	
	%M210.5	Avance	5	DIR11
	%M210.4	Petición	4	
	%M210.3	Reposo	3	
	%M210.2	Avance	2	DIR10
	%M210.1	Petición	1	
	%M210.0	Reposo	0	
%MW211	%M211.15			
	%M211.14	Avance	30	DIR19
	%M211.13	Petición	29	
	%M211.12	Reposo	28	
	%M211.11	Avance	27	DIR18
	%M211.10	Petición	26	
	%M211.9	Reposo	25	
	%M211.8	Avance	24	DIR17
	%M211.7	Petición	23	
	%M211.6	Reposo	22	
	%M211.5	Avance	21	DIR16
	%M211.4	Petición	20	
	%M211.3	Reposo	19	
	%M211.2	Avance	18	DIR15
	%M211.1	Petición	17	
	%M211.0	Reposo	16	

Tabla 6 Posiciones de memoria PLC Profibus

PLC CAN				
Dirección	Posición (bit)	Consigna	Programa InfluxDB	Campo InfluxDB
%MW212	%M212.15		15	Reserva
	%M212.14	Avance	14	DIR08
	%M212.13	Petición	13	
	%M212.12	Reposo	12	
	%M212.11	Avance	11	DIR07
	%M212.10	Petición	10	
	%M212.9	Reposo	9	
	%M212.8	Avance	8	DIR06
	%M212.7	Petición	7	
	%M212.6	Reposo	6	

	%M212.5	Avance	5	DIR05
	%M212.4	Petición	4	
	%M212.3	Reposo	3	
	%M212.2	Avance	2	DIR04
	%M212.1	Petición	1	
	%M212.0	Reposo	0	
%MW213	%M213.15			
	%M213.14			
	%M213.13			
	%M213.12			
	%M213.11			
	%M213.10			
	%M213.9			
	%M213.8	Avance	24	DIR21
	%M213.7	Petición	23	
	%M213.6	Reposo	22	
	%M213.5	Avance	21	DIR20
	%M213.4	Petición	20	
	%M213.3	Reposo	19	
	%M213.2	Avance	18	DIR09
	%M213.1	Petición	17	
	%M213.0	Reposo	16	

Tabla 7 Posiciones de memoria PLC CAN

Flujo analizador de redes:

Figura 48 Flujo acondicionamiento y posterior envió de datos del analizador

Respecto al analizador de red del trabajo "Estudio y automatización de una aplicación industrial basada en el transporte y la gestión de piezas" se ha cambiado la estructura del flujo y por ello la metodología de funcionamiento.

A mi entender se ha simplificado también el acondicionamiento de los datos para su posterior envió a la base de datos.

Se ha hecho uso de un nodo Modbus TCP para la lectura de la primera parte del analizador. “Como detalla el trabajo de Pumares Cerezo, J. es necesario el uso de dos nodos para la lectura de todos los parámetros del analizador”. Para ello en nuestro caso el primer analizador leerá los 40 primeros registros, de la Word 0 a la 40, donde cada parámetro a obtener tendrá un tamaño de 2 words, 4 bytes. Donde los bytes más significativos corresponderán a valores negativos mientras que los menos significativos a valores positivos.

Para la lectura de la segunda parte del analizador se ha hecho uso del nodo de función. Donde dicho nodo le podemos asignar mediante código de programación funciones internas semejantes al nodo Modbus TCP. Para ello buscaremos ayuda en la página oficial de NodeRed. Le asignaremos que lea los 40 registros restantes, de la word 48 a la 86. Al tratarse de un nodo de función para que no de error deberá llevar antes por defecto uno de lectura, para ello haremos uso del nodo “inject”.

El nodo de función llamado “Acondicionamiento” será el responsable de gestionar, procesar y acondicionar los datos recibidos de los nodos de lectura del analizador 1 y 2.

El programa se ha modificado con respecto al original de Pumares Cerezo. J, simplificando el proceso de operaciones y cambiando el funcionamiento haciendo uso de la función interna “topic” asociada a cada nodo de lectura.

La parte de programa para el acondicionamiento de los datos es la misma que en los PLCs, ya que como hemos comentado antes al tratarse de un nodo “Influx batch” debe segur una estructura determinada.

Los nodos de “Influx batch” y “debug” se utilizan para conseguir el mismo resultado que el anterior flujo explicado.

Para saber detalladamente cada parámetro del analizador donde está ubicado y donde se almacena en la siguiente tabla se detalla. Donde:

A. Magnitud medida [unidades]

B. Posición interna de las magnitudes del analizador en hexadecimal

C. Conversión en binario para su interpretación

D. Nombre asignación del Tópico

E. Dirección

F. Posición del byte menos significativo al más significativo

G. Posición de memoria en el programa InfluxDB

H. Nombre de los campos donde se guardara cada "measurement" [unidades]

Analizador							
A	B.	C.	D.	E.	F.	G.	H.
Tensión L1 [dV]	0	0	Analizador 1 (First)	MW0	MB1-MB0	0	V1 [V]
Corriente L1 [mA]	2	2		MW2	MB3-MB2	2	C1 [A]
Potencia Activa L1 [W]	4	4		MW4	MB5-MB4	4	PA1 [kW]
Potencia Reactiva L1 [Var]	6	6		MW6	MB7-MB6	6	PReac1 [kVar]
Factor de Potencia L1	8	8		MW8	MB9-MB8	8	FP1
Tensión L2 [dV]	A	10		MW10	MB11-MB10	10	V2 [V]
Corriente L2 [mA]	C	12		MW12	MB13-MB12	12	C2 [A]
Potencia Activa L2 [W]	E	14		MW14	MB15-MB14	14	PA2 [kW]
Potencia Reactiva L2 [Var]	10	16		MW16	MB17-MB16	16	PReac2 [kVar]
Factor de Potencia L2	12	18		MW18	MB19-MB18	18	FP2
Tensión L3 [dV]	14	20		MW20	MB21-MB20	20	V3 [V]
Corriente L3 [mA]	16	22		MW22	MB23-MB22	22	C3 [A]
Potencia Activa L3 [W]	18	24		MW24	MB25-MB24	24	PA3 [kW]
Potencia Reactiva L3 [Var]	1A	26		MW26	MB27-MB26	26	PReac3 [Var]
Factor de Potencia L3	1C	28		MW28	MB29-MB28	28	FP3
Potencia Activa III [W]	1E	30		MW30	MB31-MB30	30	PAIII [kW]
Potencia Inductiva III [Var]	20	32		MW32	MB33-MB32	32	PInducIII [kVar]
Potencia Capacitiva III [Var]	22	34		MW34	MB35-MB34	34	PCapaIII [kVar]
Cos Phi III	24	36		MW36	MB37-MB36	36	CosphiIII
Factor de Potencia III	26	38		MW38	MB39-MB38	38	FPIII

THD V1 L1 [%]	30	48	Analizador 2 (Second)	MW40	MB41-MB40	0	THDVL1 [%]
THD V2 L2 [%]	32	50		MW42	MB43-MB42	2	THDVL2 [%]
THD V3 L3 [%]	34	52		MW44	MB45-MB44	4	THDVL3 [%]
THD I L1 [%]	36	54		MW46	MB47-MB46	6	THDIL1 [%]
THD I L2 [%]	38	56		MW48	MB49-MB48	8	THDIL2 [%]
THD I L3 [%]	3A	58		MW50	MB51-MB50	10	THDIL3 [%]
Energía Activa III [Wh]	3C	60		MW52	MB53-MB52	12	EnergActiIII [kWh]
Energía reactiva Inductiva [Varh]	3E	62		MW54	MB55-MB54	14	EnReacInd [kVarh]
Energía reactiva Capacitiva [Varh]	40	64		MW56	MB57-MB56	16	EnReacCap [kVarh]
Potencia Aparente III [VA]	42	66		MW58	MB59-MB58	18	PAIII [kVA]
Corriente Neutro [mA]	48	72		MW64	MB65-MB64	24	CorrNeutro [A]
Potencia Aparente L1 [VA]	4A	74		MW66	MB67-MB66	26	PA11 [kVA]
Potencia Aparente L2 [VA]	4C	76		MW68	MB69-MB68	28	PA22 [kVA]
Potencia Aparente L3 [VA]	4E	78		MW70	MB71-MB70	30	PA33 [kVA]
Temperatura [dº]	50	80		MW72	MB73-MB72	32	Temp [º]
Energía Aparente III [VAh]	56	86		MW78	MB79-MB78	38	EnergApIII [kVAh]

Tabla 8 Posiciones de memoria analizador

4.1.2 Almacenamiento en el servidor local (InfluxDB)

Una vez acondicionados estos datos, gracias al software de NodeRed atravesé de su nodo interno "Influx batch" (con la misma dirección que la base de datos), se almacenan en orden cronológico de manera serial teniendo en cuenta el valor temporal en que se recoge dicho dato, obteniendo una lista de parámetros del caso estudio, llamados campos, "fields" ubicados en columnas y estructurada, como hemos dicho antes, en orden histórico.

Dichos parámetros obtenidos de la celda estarán agrupados dentro de los "measurements" que a su vez estarán dentro de la base de datos creada.

Estructura creada para la base de datos:

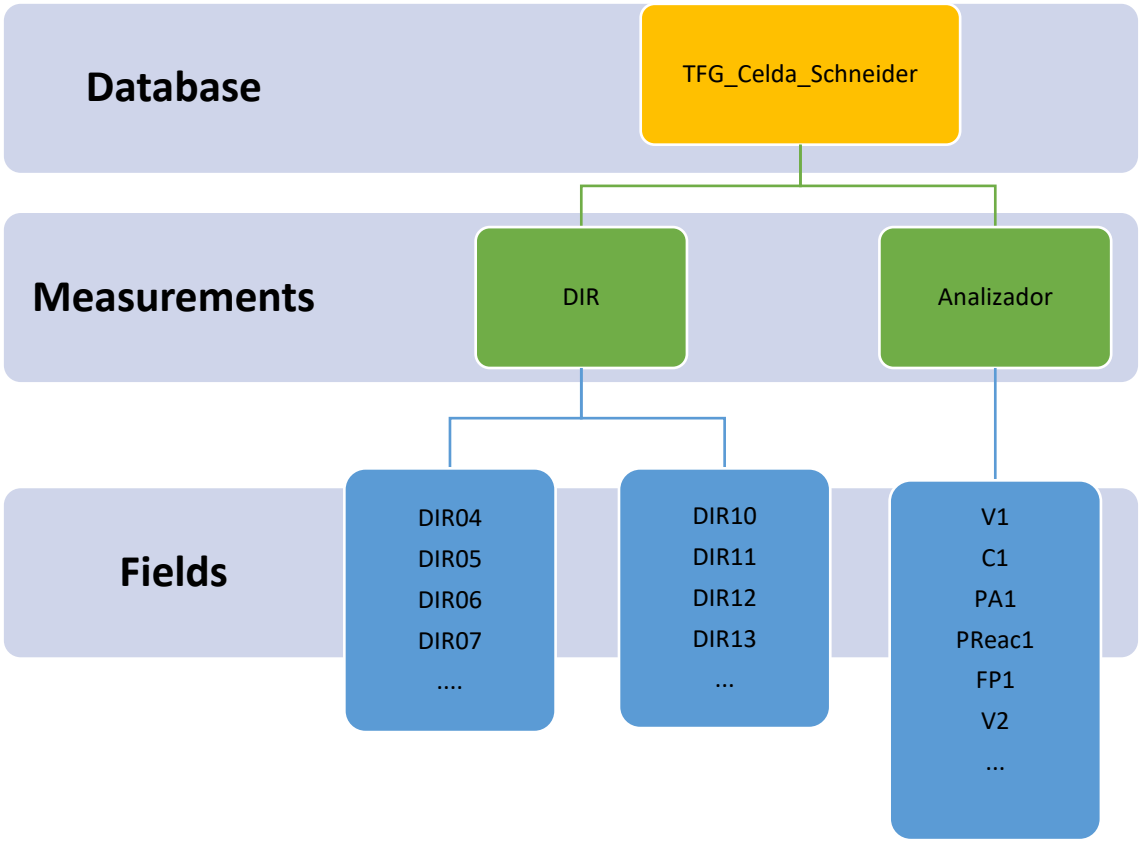


Figura 49 Estructura Base de datos

En la figura de abajo podemos ver representada la estructura interna del programa. En este caso hemos llamado los datos almacenados de los retenedores.

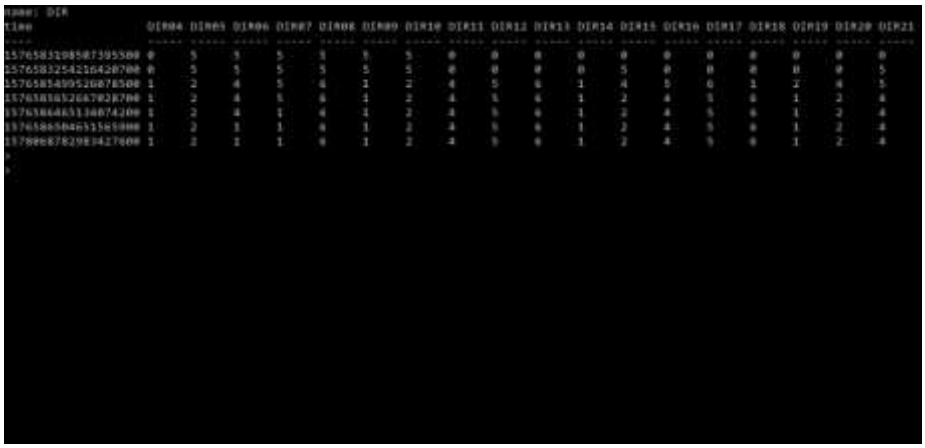


Figura 50 Base de datos InfluxDB retenedores

4.1.3 Visualización (Grafana)

Con las tablas de datos almacenadas cronológicamente dentro de nuestra base de datos ya podremos visualizar dichas tablas de cualquier manera haciendo uso de los diferentes "dashboards" que vienen por defecto con la descarga del Grafana.

Antes de empezar la creación del tablero de instrumentos "dasboard" deberemos añadir la dirección de la base de datos como pasaba con NodeRed.

De los distintos tipos de "dasboards" que tenemos por defecto, de origen, no venía ninguno idóneo para la representación del caso estudio. Fue necesario descargar nuevos "plugins" des de la página web oficial.

4.1.4.1 Plugin Imagelt

Esta extensión, complemento permite superponer pantallas de medición sobre una imagen web.

Estas pantallas de medición les llamaremos sensores y les podremos asignar un color , cambiar su tamaño y serán fácilmente arrastrables permaneciendo en la posición donde la ubiques de la imagen, cambiándolo otra vez de posición solo si abres el candado para poderlo editar.

Figura 51 Logo plugin Imagelt

4.1.4.2 Dashboard retenedores

Como hemos comentado en el apartado 4.1.2, el estado de los retenedores se va almacenando en la base de datos creada. Cada vez que se actualiza la página del Grafana nos mostrara el último registro que se ha almacenado en el servidor.

Figura 52 Dashboard celda caso estudio de los retenedores

4.1.4.3 Dashboard Analizador de redes

Como no podía faltar también se ha procedido a visualizar los parámetros del analizador de redes mediante el uso de todo tipo de "dasboards".

El que visualiza mejor el valor del parámetro y de forma gráfica en el transcurso del tiempo es el "graph". Con el podremos interpretar fácilmente los consumos diarios y mejorar su funcionamiento energéticamente hablando.

Visualización de las tensiones e intensidades así como las diferentes potencias:

Figura 53 Dashboard analizador (tensiones, intensidades, potencias)

Los THD, energía consumida e incluso la temperatura dentro del armario:

Figura 54 Dashboard analizador (THDV, THDI, energías, corriente neutro y temperatura)

Los factores de potencia y cos de phi:

Figura 55 Dashboard analizador (potencias III, cosphi III y factor de potencia III

CAPÍTULO 5. VALIDACIÓN

5.1 Pruebas funcionales

5.1.1 Envió de datos del PLC al visualizador Grafana

Para ello debemos construir el flujo de la figura 47 para la lectura de los PLCs.

Para cada nodo de entrada "modbustcp" se tendrá que asignar la dirección IP de cada host (PLC), la dirección interna del PLC a la que se quiere acceder junto a su tamaño y el tipo de dato enviado.

Sabiendo previamente el estado en que se cogerán los datos se procederá a su acondicionamiento.

Al tratarse de datos de estado de los retenedores, se han realizado dos programas que tienen resultados parecidos pero que funcionan diferente.

El primer programa se ha realizado teniendo en cuenta que los datos están recogidos en marcas, leeremos booleanos con el código de función FC1.

```
1  bpl=msg.payload;
2
3  for(i=0;i<54;i=i+3){
4  if(opl[i]===true&&opl[i+1]===false&&opl[i+2]===false){opl[i]=1;}//Reposo
5  if(opl[i]===false&&opl[i+1]===true&&opl[i+2]===false){opl[i]=2;}//Peticion
6  if(opl[i]===false&&opl[i+1]===false&&opl[i+2]===true){opl[i]=4;}//Avance
7  if(opl[i]===true&&opl[i+1]===false&&opl[i+2]===true){opl[i]=5;}//Reposo+Avance
8  if(opl[i]===false&&opl[i+1]===true&&opl[i+2]===true){opl[i]=6;}//Peticion+Avance
9  }
```

Figura 56 Código de programa lectura de booleanos

Ver todo el código del programa en el anexo VII.

El segundo programa se ha realizado teniendo en cuenta que los datos están recogidos en registros. Cada registro es una Word de 16 bits, leeremos words con el código de función FC3.

En un primer momento la idea era utilizar dos bucles, uno para la lectura de cada Word y otro para la lectura de cada posición de memoria. Pero no se encontraba la manera de declarar la variable que almacenaría ese resultado.

```
var coil=[],[];
opl=msg.payload;

    for(i=0;i<3;i=i++){
    for(j=0;j<16;j++){
        if((opl[i]&Math.pow(2,j))>0){coil[i][j]=1}

    }
}
msg.payload=[
```

Figura 57 Código de programa del bucle de lectura de registros (words) y posiciones (bits)

En lugar de eso se optó por eliminar el bucle de lectura de cada word a expensas de tener líneas de código repetitivas.

```
for(j=0;j<16;j++){
    if((opl0[1]&Math.pow(2,j))>0){opl[j]=true}
    else {coil[j]=false}
     for(i=0;i<16;i=i+3){
                if(opl[i]===true&&opl[i+1]===false&&opl[i+2]===false){coil[i]=1;}//Reposo
                if(opl[i]===false&&opl[i+1]===true&&opl[i+2]===false){coil[i]=2;}//Peticion
                if(opl[i]===false&&opl[i+1]===false&&opl[i+2]===true){coil[i]=4;}//Avance
                if(opl[i]===true&&opl[i+1]===false&&opl[i+2]===true){coil[i]=5;}//Reposo+Avance
                if(opl[i]===false&&opl[i+1]===true&&opl[i+2]===true){coil[i]=6;}//Peticion+Avance
     }
}
```

Figura 58 Código de programa de lectura del primer registro

El código de la figura 58 está hecho para la lectura de tres words.

Ver todo el código del programa en el anexo VIII.

5.1.2 Envió de datos del analizador al visualizador Grafana

Construir el flujo de la figura 48 para la lectura del analizador.

Para el nodo de entrada "modbustcp" se tendrá que asignar la dirección IP de la pasarela, la dirección interna del analizador a la que se quiere acceder junto a su tamaño y el tipo de dato enviado (en este caso en FC3 para leer words).

Tendiendo el nodo “modbustcp” configurado se procederá a su acondicionamiento.

El programa creado para su correcto envió e interpretación se puede ver en la figura 59.

Ver todo el código de programa ver anexo IX

```
1
2  if(msg.topic=="First"){
3      opl=msg.payload;
4      for(i=0;i<40;i=i+2){
5
6      if(opl[i]>32767){
7
8          opl[i]=-Math.pow(2,32)+(opl[i]*Math.pow(2,16))+opl[i+1];
9          opl[i+1]=0;
10     }
11     else{
12         opl[i]=opl[i]*Math.pow(2,16)+opl[i+1];
13         opl[i+1]=0;
14
15     }
16 }
```

Figura 59 Código de programa analizador

5.2 Pruebas operacionales

5.2.1 Prueba 1 (formato de envió)

Descripción

Enviar a la base de datos utilizando el nodo “inject” de NodeRed configurado en distintas formas, un valor booleano (true, false), valor numérico cualquiera y una vector.

Primero para ello haremos uso del nodo “debug” para su visualización interna dentro del programa y poder entender de manera practica el funcionamiento de los flujos.

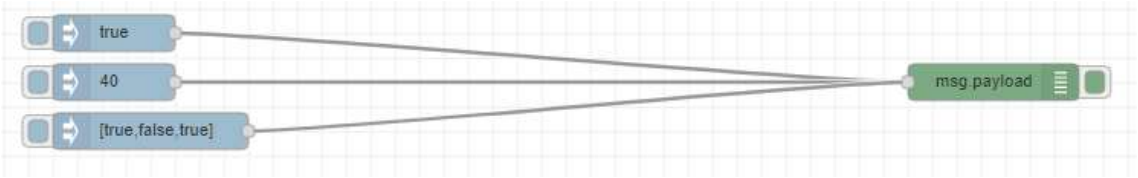


Figura 60 Ejemplo flujo. Adquisición booleanos y vectores.

Resultado esperado

Se espera que devuelva los mismos valores inscritos en el nodo de inyección.

Resultados

Como resultado efectivamente visualizamos los mismos valores sin modificar y en orden. Del bit menos significativo al más significativo en el caso de la vector.

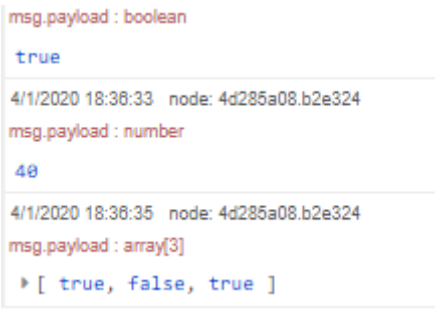


Figura 61 Verificación de recepción de datos interna en NodeRed

5.2.2 Prueba 2 (link base de datos)

Descripción

Enviar una vector a un nodo de "influx bach" direccionado a nuestra base de datos.

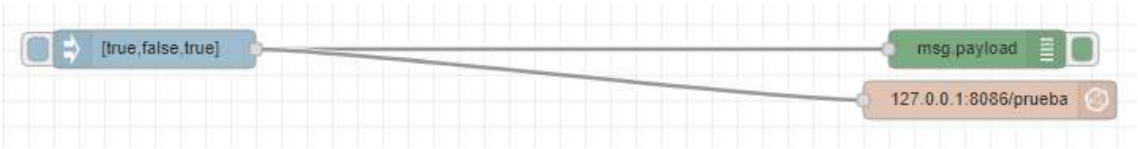


Figura 62 Ejemplo flujo. Envió de un vector a la base de datos

Resultado esperado

Visualizar dichos valores dentro de la base de datos.

Resultados

En el nodo "debug" interno del NodeRed se recibe correctamente la información pero el nodo de "influx buch" aparece un error. Nos dice que no se puede analizar, que no están definidos los campos.

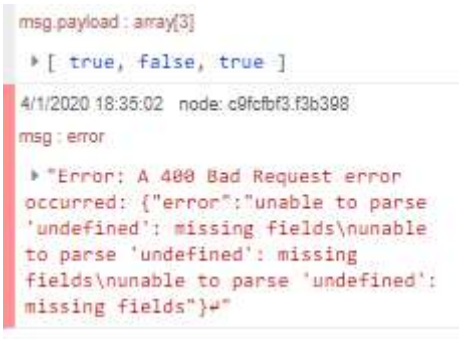


Figura 63 Visualizando errores en la recepción de datos por parte de servidor.

Modificación

Entendemos que la forma en que se envía dicha información al nodo de "Influx buch" no es fácilmente leído y reconocible. Buscaremos en la guía de NodeRed y en concreto el uso del nodo "Influx buch" como se debe procesar y enviar los datos.

Hay dos tipos de nodos de salida para InfluxDB, los dos nos exigen que cumplamos con una estructura en concreto para acondicionar los datos. Dicha estructura se deberá escribir en código java de programación utilizando el nodo de función que permite su escritura.

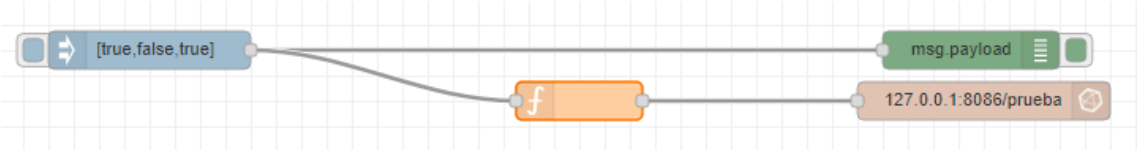


Figura 64 Ejemplo flujo. Envió de una vector al servidor utilizando un nodo de función

Rectificación

Se ha eliminado el error y se puede ver en la base de datos el vector formado por tres booleanos.

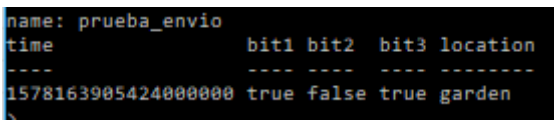


Figura 65 Comprobación de la información enviada abriendo la base de datos.

5.2.3 Prueba 3 (link modbustcp)

Descripción

Utilizar un nodo "modbustcp" de entrada para leer datos de memoria del PLC. En este caso práctico de estudio se utilizara la dirección del PLC CAN con el código de función en FC1.

Figura 66 Ejemplo flujo. Adquisición de datos utilizando el nodo de modbustcp

Resultado esperado

Comprobar que se reciben los datos en NodeRed.

Resultados

Efectivamente estamos recibiendo valores true o false del PLC.

Modificación

Ahora cambiaremos el código de función para leer registros, con el código de función FC3.

Rectificación

Visualizamos los resultados en forma de vector.

5.2.4 Prueba 4 (primeros dashboards)

Descripción

Con la base de datos creada (llamada prueba) visualizar en nuestro "dashboard" el último valor del campo (M1) de nuestro measurement (CAN).

Resultado esperado

Obtener por pantalla el número gráfico.

Resultados

Figura 67 Dashboard Gauge

Para ello se ha tenido que cambiar dentro del visualizador "Gauge" en la ventana de "display" la opción "mean" por "last value". Recomendable en la pestaña de "query" quitar las funciones "mean()" del "select" y "time intervals()" del "Grup by". No son necesarias.

Modificación

Queremos enviar todos los valores registrados y en función del tiempo.

Rectificación

Hay que seleccionar el visualizador "Graph" (para ver el valor de los datos en función del tiempo) y en caso de que tengamos un registro amplio de valores guardados hacer uso de la función "time intervals" dentro de las "querys". Nos ayudara a optimizar el grafico.

5.2.5 Prueba 5 (superponer objetos en Grafana)

Descripción

Conseguir superponer en una imagen cualquier objeto modificable que puedan representar estados o información del PLC.

Resultado esperado

Se espera conseguir que un valor introducido en NodeRed se visualice encima de una imagen descargada de una página web.

Resultados

Con "Picturel" podemos poner los objetos llamados "sensores" mediante coordenadas encima de la imagen, pero solo muestran un valor numérico. Nos interesa poder asignar colores que representen el funcionamiento o estados diferentes.

Modificación

Utilizamos otra herramienta de visualización llamada "Imagelt".

Rectificación

Esta herramienta es más fácil e intuitiva de utilizar y te permite asignar colores a mediante condiciones.

CAPÍTULO 6. CONCLUSIONES

6.1 Resumen y conclusiones del trabajo

Como en todos los servicios de tecnologías de la información y comunicación en este trabajo hemos podido comprobar que están conformados por una serie de capas o componentes que configuran una arquitectura y a su vez constan de una serie de tecnologías, servicios y protocolos.

Esta estructura diseñada es la que utilizaremos para crear el internet de las cosas, un sistema de automatización propio de la industria 4.0.

El objetivo principal del caso estudio era la recolección (ingesta) de datos, la transmisión de estos datos, (redes de comunicación, conectividad), el almacenamiento de dichos datos en un centro de datos (servidor local o en la nube), posterior analítica de estos datos y finalmente su presentación y visualización de resultados.

Para hacer posible todo esto es necesario un conocimiento previo ya que existen en el mercado diferentes productos tecnológicos para dicho fin. Al ahora de elegir uno o el otro tendremos en cuenta muchos factores como por ejemplo el entorno de trabajo, la distancia entre el conexionado (distancia máxima del primer al último nodo de la red), la velocidad a la que se transmite los datos en bits/s y el número de nodos o elementos máximos que se puedan conectar a la red o al bus entre otros.

En conclusión las empresas de hoy en día tienen que adoptar esta metodología e implementarla, digitalizándola. Con ello conseguiremos conectar las empresas a las redes, mejorar su eficiencia, gestionar los activos de forma más sostenible ayudando al medio amiente.

Según todos los pronósticos los asentamientos tecnológicos dados en el año 2016 serán los que marquen el porvenir de los próximos acontecimientos.

Estos tres pilares son el Big Data (para nosotros la base de datos) y su analítica de estos datos (tendencia más consolidada y solicitada en organizaciones y empresas), el internet de las cosas, los llamados "IoT" (los miles de millones de objetos conectados entre sí a través de internet), y por último la inteligencia artificial aplicada, un pilar que ha pasado a ser una tecnología emergente des de sus dos tecnologías más significativas des del punto de vista del mercado y los negocios: aprendizaje automática y aprendizaje profundo.

Lo que hemos llevado a cabo en este trabajo es implementar los dos primeros pilares, entender la estructura de su hardware y software, y estudiar el funcionamiento en que se rigen para llevarlo a cabo.

BIBLIOGRAFÍA

Forouzan, B. (2007). Transmisión de datos y redes de comunicaciones. Madrid: Mc-graw Hill.

Joyanes, L. (2017). Industría 4.0. La cuarta revolución industrial. México: Macrombro.

Martínez Ruiz, P. Estudio de las etapas de diseño e implementación de una arquitectura de supervisión para procesos industriales. Trabajo de fin de máster, UPC, 2019.

Pumares Cerezo, J. Estudio y automatización de una aplicación industrial basada en el transporte y la gestión de piezas. Trabajo de fin de grado, UPC, 2019.

Tanenbaum, A. y Wetherall, D. (2012). Redes de computadores. Mèxico: Pearson.

WEBGRAFÍA

ABB. Comunicaciones Industriales: Conceptos básicos. (2018). Recuperado en https://www.youtube.com/watch?v=SU_yM50tz6E&t=310s&fbclid=IwAR0bhLFtb5353VNmiH4O5EIFEeMwHBJDQSvLejUWNmWl0ucBbnByjWedAqU

Blanco, R. Fontrodona, J. i Poveda C. (2016) a partir de datos de Smit et al. (2016). LA INDUSTRIA 4.0: EL ESTADO DE LA CUESTIÓN. https://www.mincotur.gob.es/Publicaciones/Publicacionesperiodicas/EconomiaIndustrial/RevistaEconomiaIndustrial/406/BLANCO,%20FONTRODONA%20Y%20POVEDA.pdf

Codificación Manchester. Wikipedia. Recuperado en https://es.wikipedia.org/wiki/Codificaci%C3%B3n_Manchester

CISCO. TCP vs UDP. Recuperado en https://sites.google.com/site/ciscoch9/home/tcp-vs-udp

Diffen. TCP vs UDP. Recuperado en https://es.diffen.com/tecnologia/TCP-vs-UDP

El futuro es ahora: la industria 4.0. (2019). Recuperado en https://www.escueladenegocio.com/blog/industria-4-0/

Estructura interfaz de comunicación maestro y esclavo. Recuperado en https://prezi.com/iyg7zmnlztwk/estructura-interfaz-de-comunicacion-maestro-y-esclavo/

Ethernet. Wikipedia. Recuperado en https://es.wikipedia.org/wiki/Ethernet

Formato de la trama Ethernet. Recuperado en http://redesdecomputadores.umh.es/enlace/ethernet/Formato_Trama_ethernet.html

Gómez, E. Cómo funciona el Puerto Serie y la UART. (2017). Recuperado en https://www.rinconingenieril.es/?fbclid=IwAR1Ii0LmZAJ5Vv7gxVhYViaEqV02JTgsvw9HbnqSsf3h5hy448oDxsMrtGY

Grafana. Descarga del programa Grafana. Recuperado en https://grafana.com/docs/guides/getting_started/

Grafana. Plugin Imagelt. 2019. Recuperado en https://grafana.com/grafana/plugins/pierosavi-imageit-panel

Influx. Descarga del programa InfluxDB. Recuperado en https://portal.influxdata.com/downloads/

Node-Red. Descarga del programa Node-RED. Recuperado en https://nodered.org/docs/getting-started/local

Node-Red. node-red-contrib-influxdb. 2019. Recuperado en https://flows.nodered.org/node/node-red-contrib-influxdb

Node-Red. node-red-contrib-modbustcp. 2019. Recuperado en https://flows.nodered.org/node/node-red-contrib-modbus

Manual de planificación e instalación del sistema advantys STB. (2009). Recuperado en

https://www.se.com/es/es/download/document/31002950K01000/

Manual autómatas Modicon Premium (2005). Recuperado en

https://e-archivo.uc3m.es/bitstream/handle/10016/8033/Manuales.pdf

Manual Modicon M340 (2018). Recuperado en

https://www.se.com/es/es/download/document/35015195K01000/

Power Cert. CompTIA Network + Certification Video Course. (2017). Recuperado en https://www.youtube.com/watch?v=vrh0epPAC5w

Reto 5: Comunicándonos con arduino puerto serie. Recuperado en

https://educarparaelcambio.com/arduino/reto-5-comunicandonos-con-arduino-puerto-serie/

Vergara, K. (6 de mayo de 2007). Topología de red: malla, estrella, árbol, bus y anillo. Recuperado en https://mimateriaenlinea.unidedu.mx/dts_cursos_mdl/lic/TIC/IT/AM/03/Topologia.pdf

Seneca, Z-GATEWAY, ModBUS RTU<>ModBUS TCP/IP Gateway, Dual RS485 port. Recuperado en

https://www.microlectra.nl/p-seneca-z-gateway-modbus-rtultgtmodbus-tcpip-gateway-dual-rs485-port-5495